US006448168B1

(12) **United States Patent**
Rao et al.

(10) **Patent No.: US 6,448,168 B1**
(45) **Date of Patent: Sep. 10, 2002**

(54) **METHOD FOR DISTRIBUTING A CLOCK ON THE SILICON BACKSIDE OF AN INTEGRATED CIRCUIT**

(75) Inventors: **Valluri R. Rao**, Saratoga; **Jeffrey K. Greason**, Tehachapi; **Richard H. Livengood**, Los Gatos, all of CA (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/483,573**

(22) Filed: **Apr. 14, 2000**

Related U.S. Application Data

(62) Division of application No. 08/938,486, filed on Sep. 30, 1997, now Pat. No. 6,037,822.

(51) **Int. Cl.**[7] **H01L 21/44**; G06F 1/04; G01R 31/08

(52) **U.S. Cl.** **438/598**; 438/668; 438/622; 327/298; 370/252

(58) **Field of Search** 438/598, 599, 438/597, 667, 668, 622; 327/298, 565, 293, 262; 326/101, 39, 95; 370/252

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,659,968	A *	4/1987	Nakamura	438/668
4,731,781	A *	3/1988	Nakamura	438/668
5,498,755	A *	3/1996	Bayraktaroglu	438/667
5,501,006	A *	3/1996	Gehman, Jr. et al.	29/840
5,537,498	A *	7/1996	Bausman et al.	327/293
5,614,442	A *	3/1997	Tserng	438/667
5,717,229	A *	2/1998	Zhu	257/208
5,760,478	A *	6/1998	Bozso et al.	257/777
6,157,237	A *	12/2000	Mitra	327/295

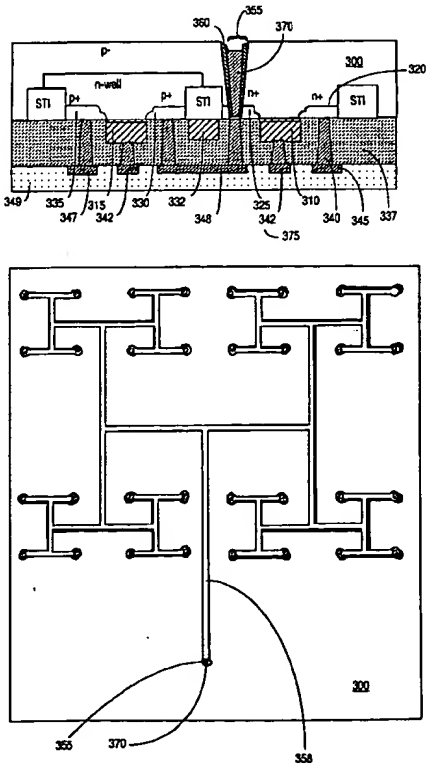
* cited by examiner

Primary Examiner—Keith Christianson
Assistant Examiner—Yennhu B. Huynh
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

A method and apparatus for clocking an integrated circuit. The apparatus includes an integrated circuit having a clock driver disposed in a first side of a semiconductor substrate, and a clock distribution network coupled to the clock driver and disposed in a second side of the semiconductor substrate to send a clock signal to clock an area of the integrated circuit.

13 Claims, 12 Drawing Sheets



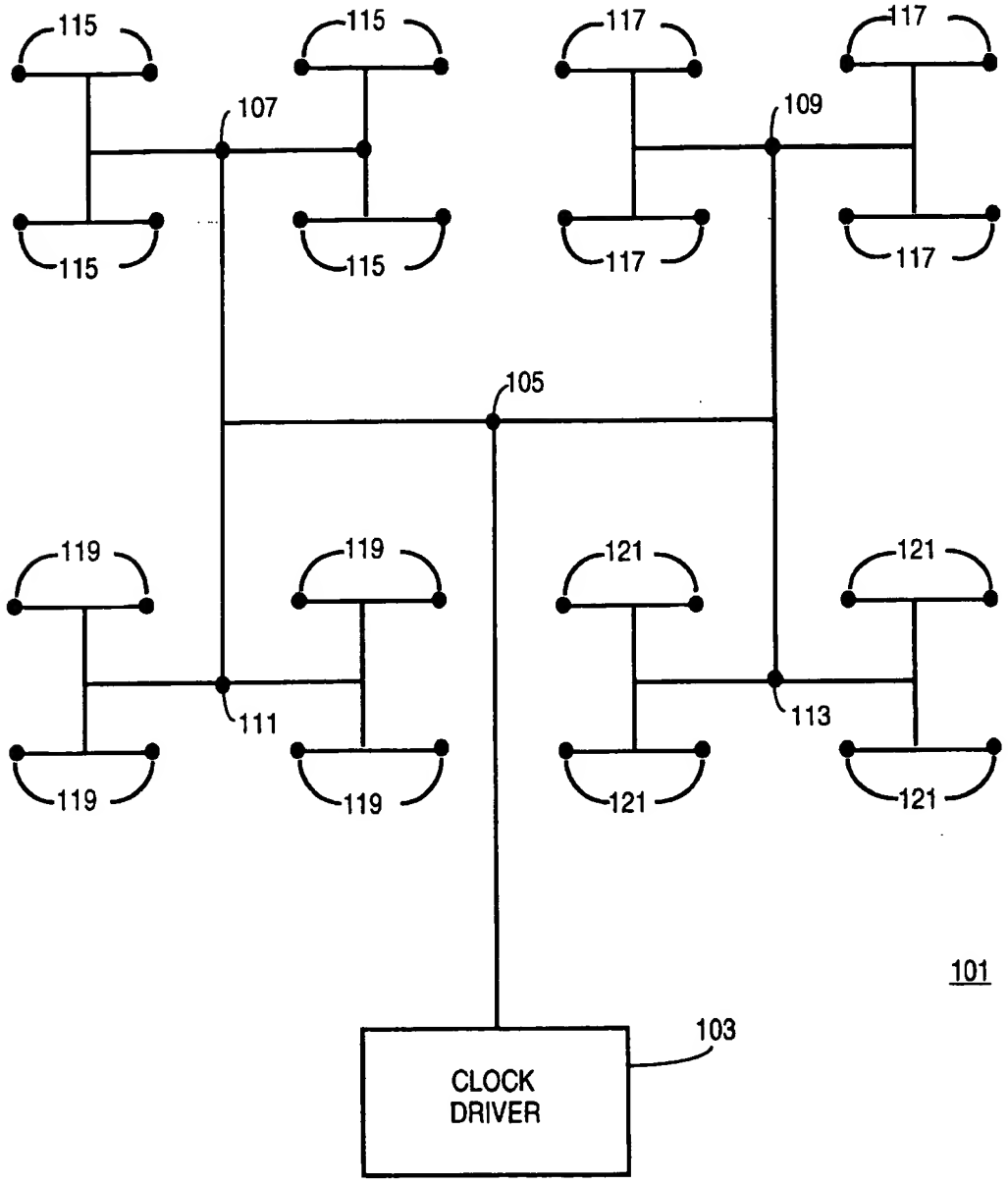


FIG. 1
(PRIOR ART)

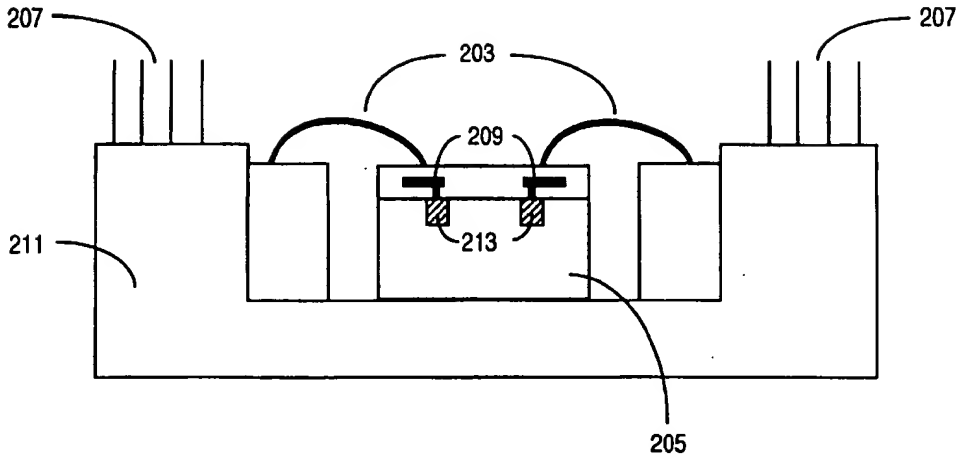


FIG. 2
(PRIOR ART)

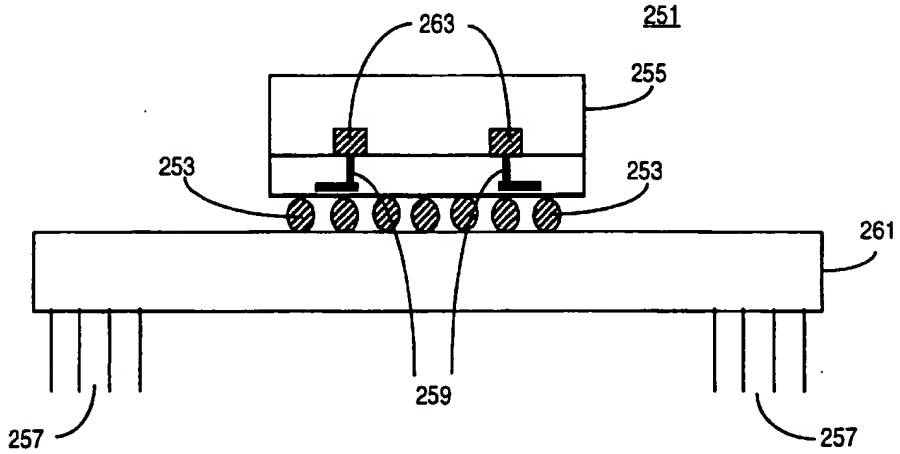


FIG. 3
(PRIOR ART)

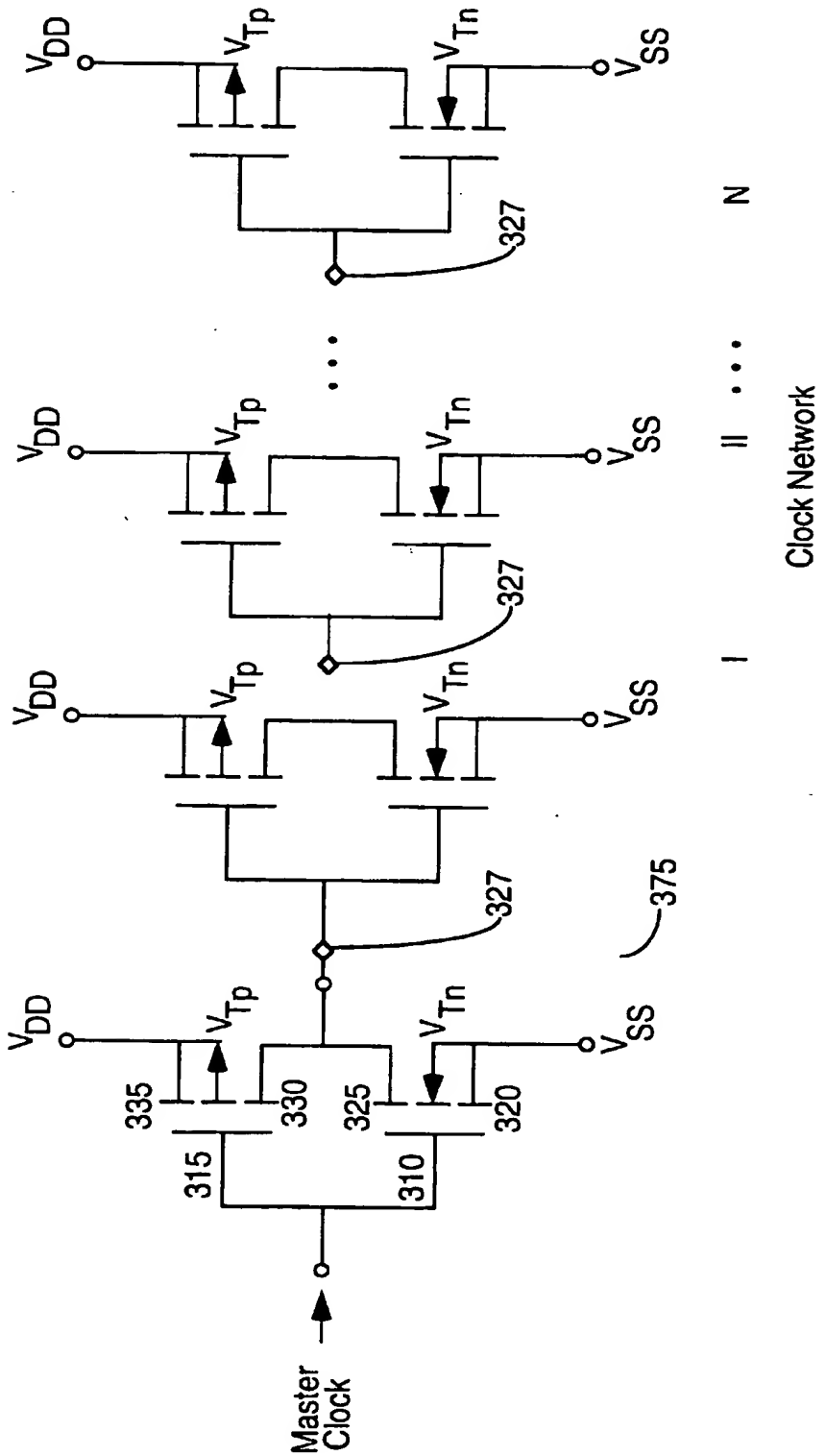

V_{DD}
V_{Tp}
V_{Tn}
V_{SS}
327
375
335
330
325
320
315
310
Master Clock
N
I
II
Clock Network

FIG. 4

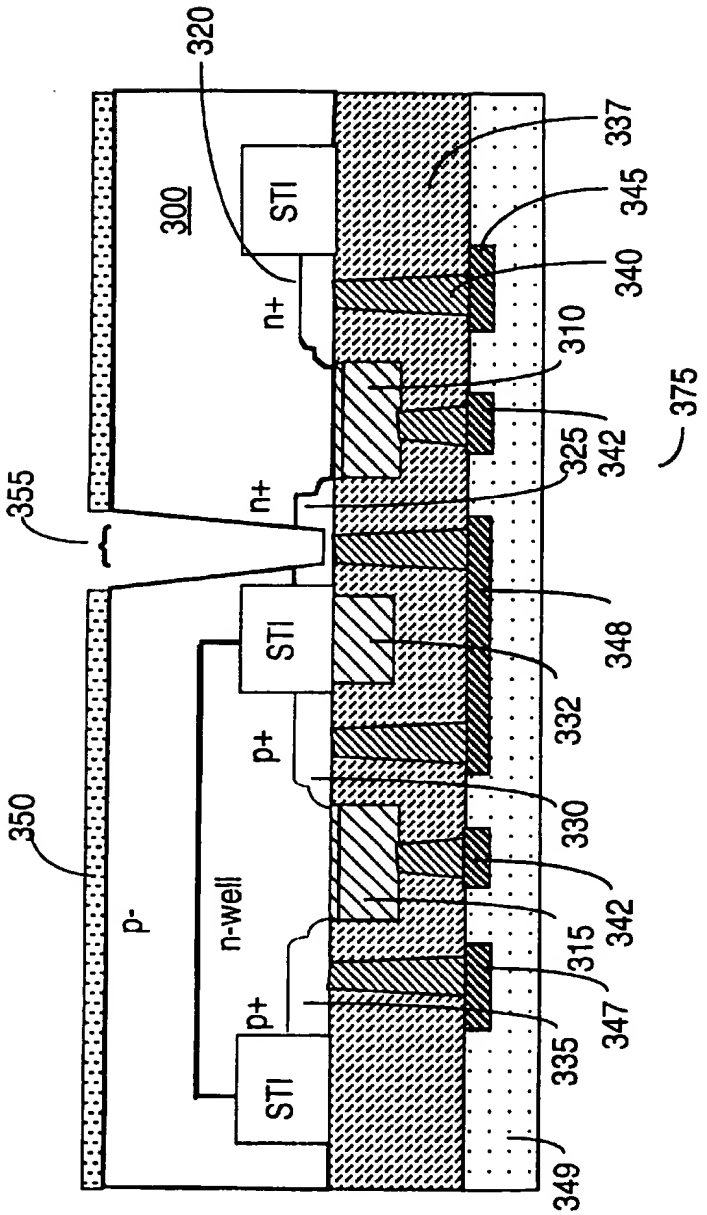

320
300
STI
n+
337
345
340
310
375
325
342
355
n+
348
STI
332
p+
330
350
p-
n-well
342
315
p+
347
335
STI
349

FIG. 5

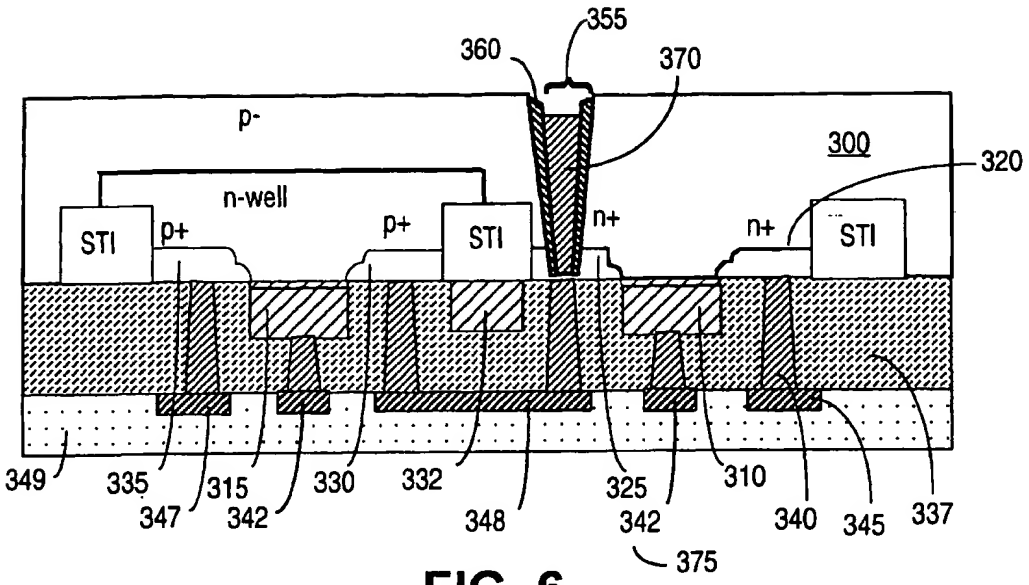

355
360
370
p-
300
320
n-well
STI
p+
p+
STI
n+
n+
STI
349
335
347
315
342
330
332
348
325
342
310
340
345
337
375

FIG. 6

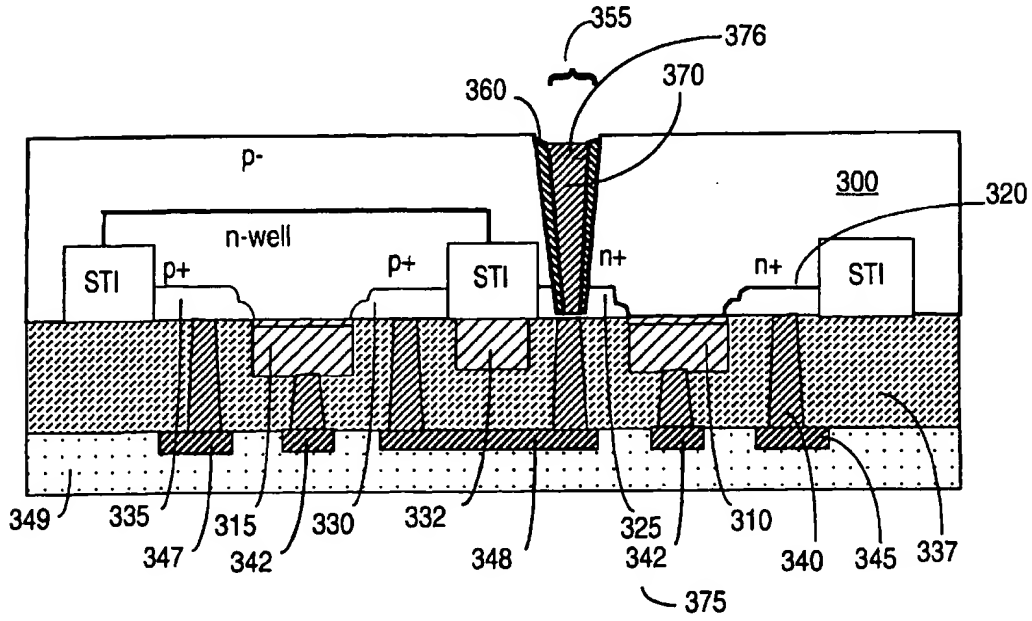

355
376
360
370
p-
300
320
n-well
STI
p+
p+
STI
n+
n+
STI
349
335
347
315
342
330
332
348
325
342
310
340
345
337
375

FIG. 7

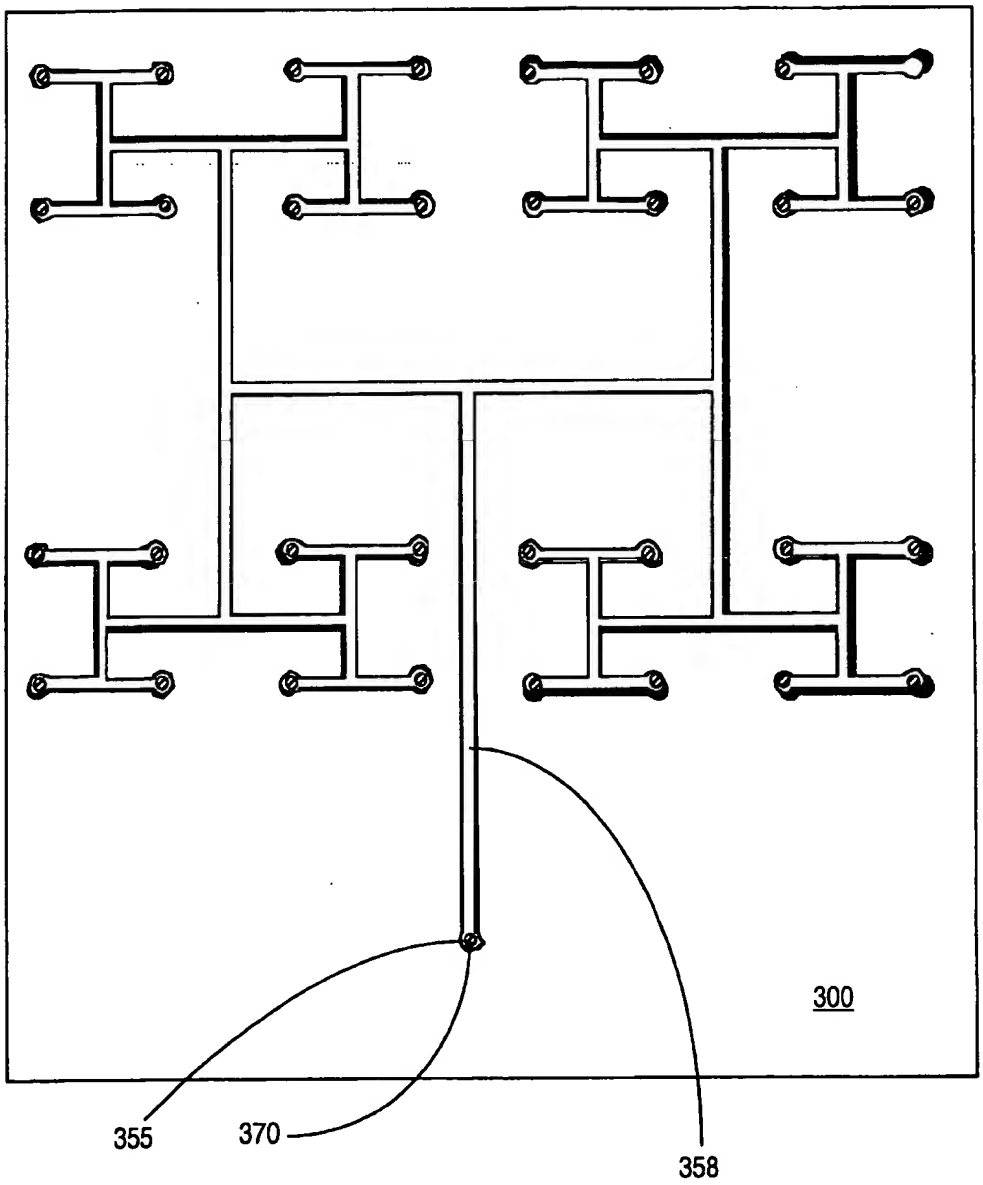


FIG. 8

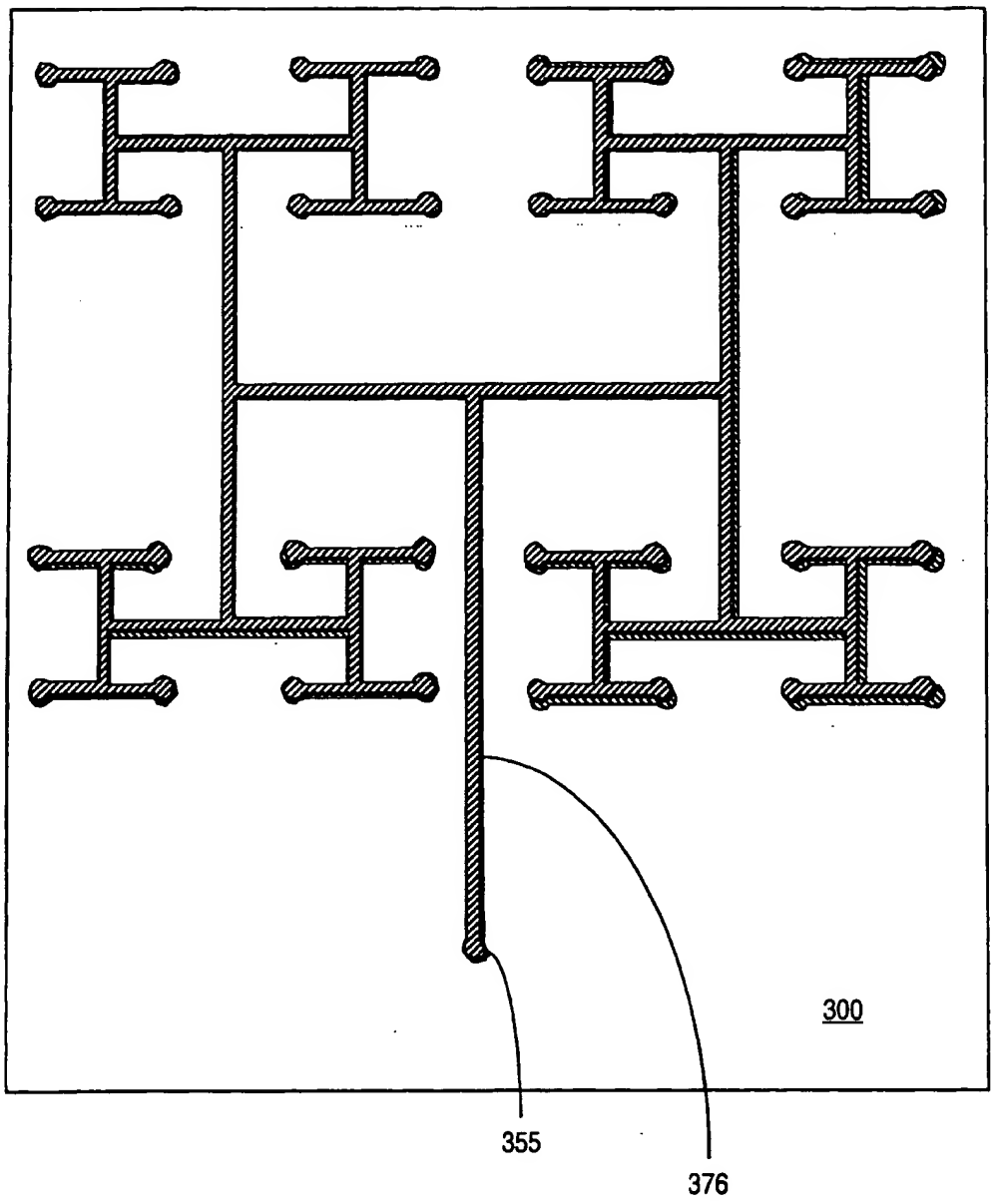


FIG. 9

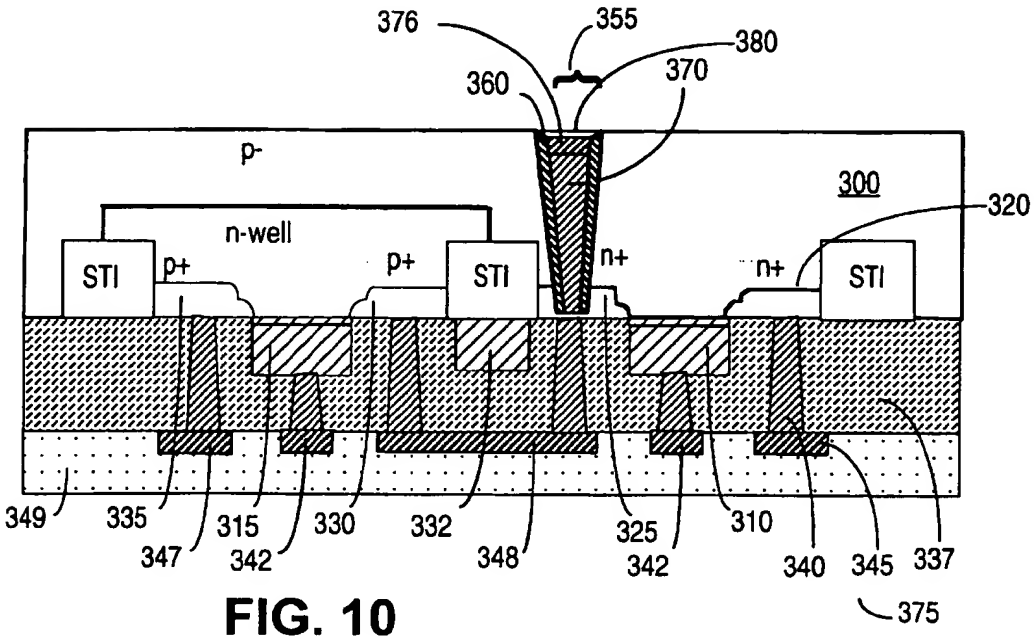

376
355
380
360
370
p-
300
320
n-well
STI
p+
p+
STI
n+
n+
STI
349
335
347
315
342
330
332
348
325
342
310
340
345
337
375

FIG. 10

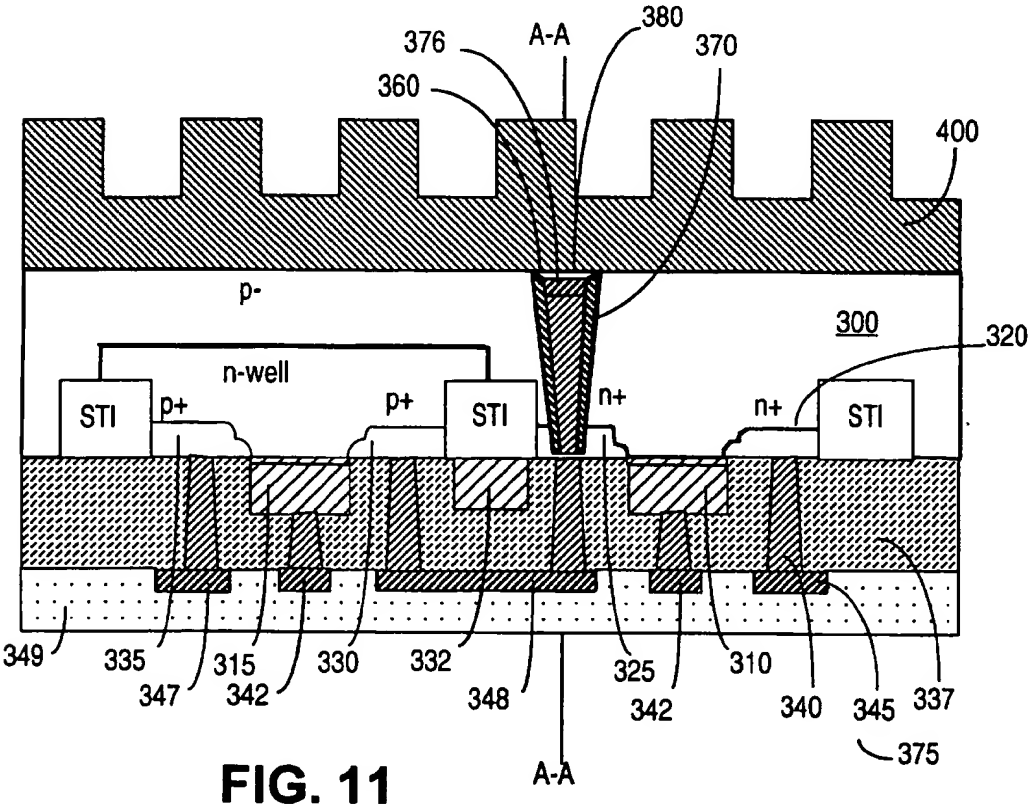

376
A-A
380
370
360
400
p-
300
320
n-well
STI
p+
p+
STI
n+
n+
STI
349
335
347
315
342
330
332
348
325
342
310
340
345
337
375
A-A

FIG. 11

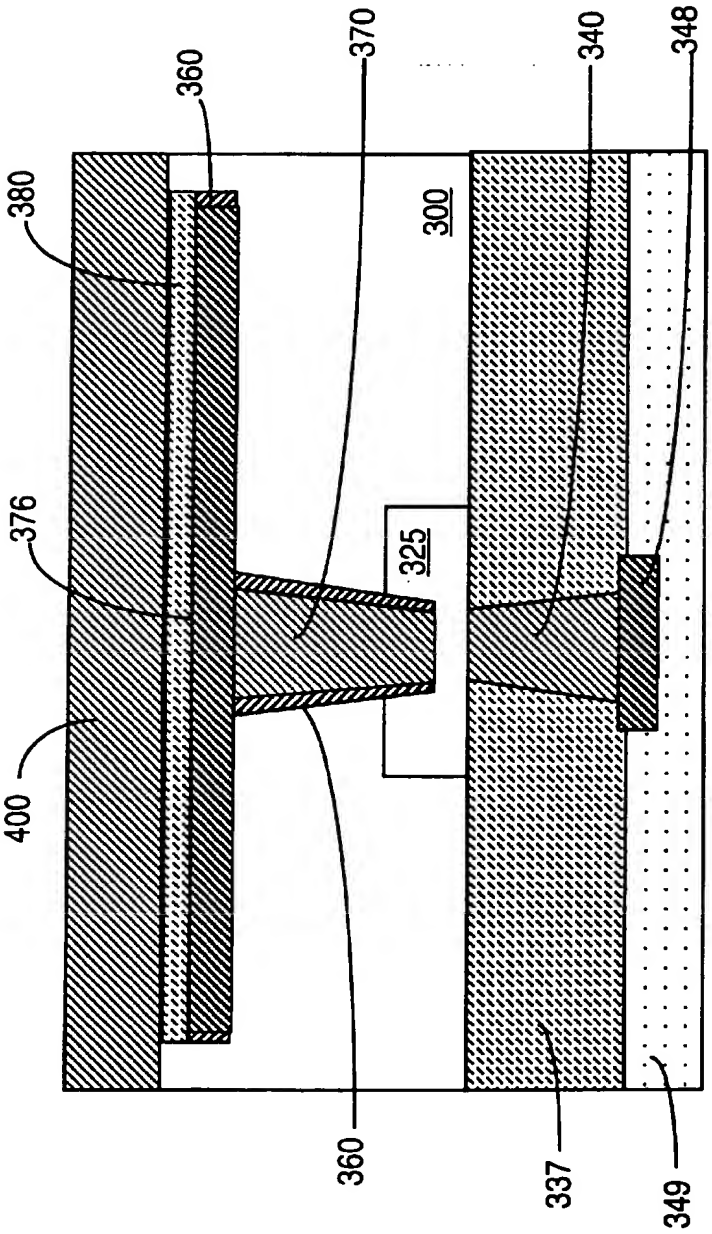


FIG. 12

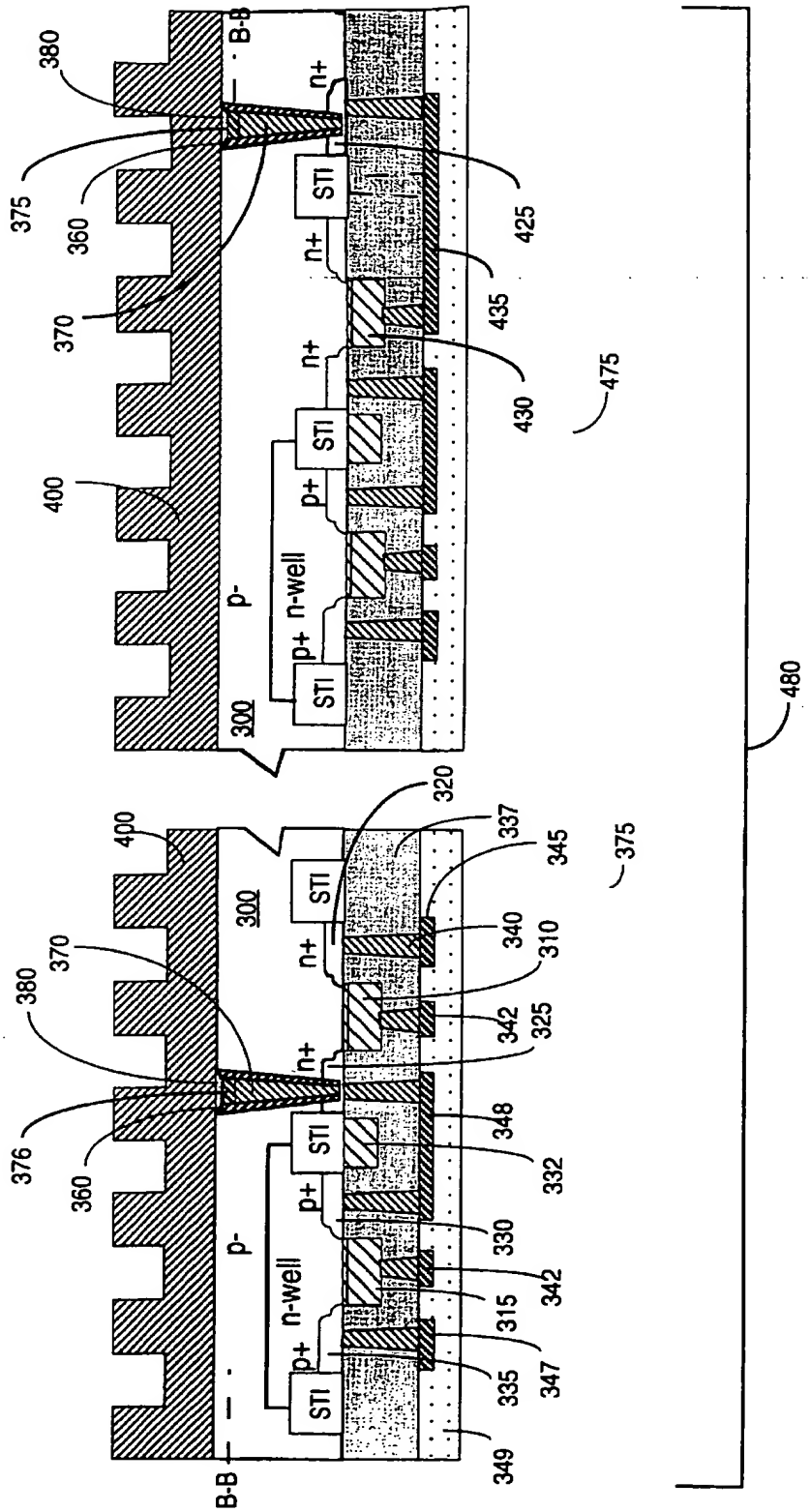

380
375
360
370
400
B-B
p-
n-well
300
STI
n+
p+
425
435
430
475
480
320
337
345
375
340
310
342
325
348
332
330
315
335
347
349
376

FIG. 13

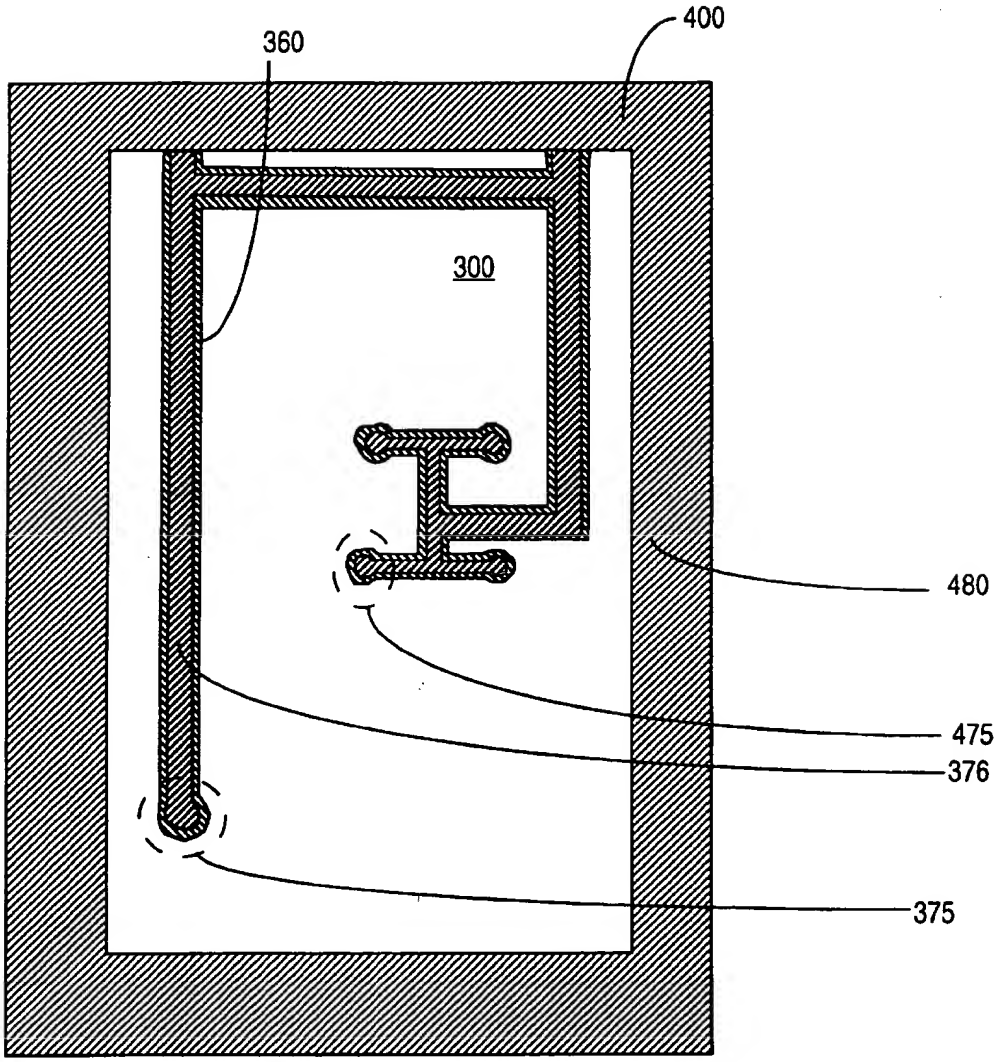


FIG. 14

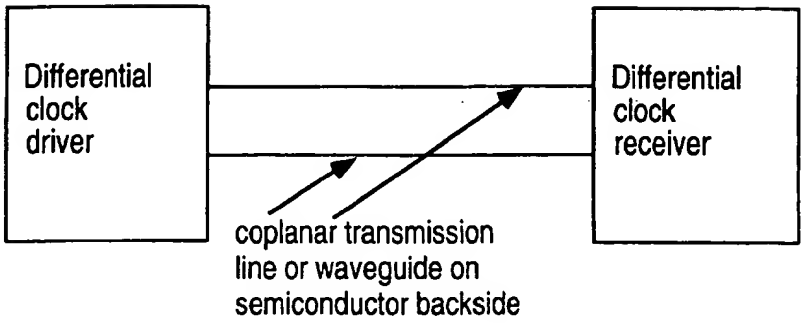

Differential clock driver
Differential clock receiver
coplanar transmission line or waveguide on semiconductor backside

FIG. 15

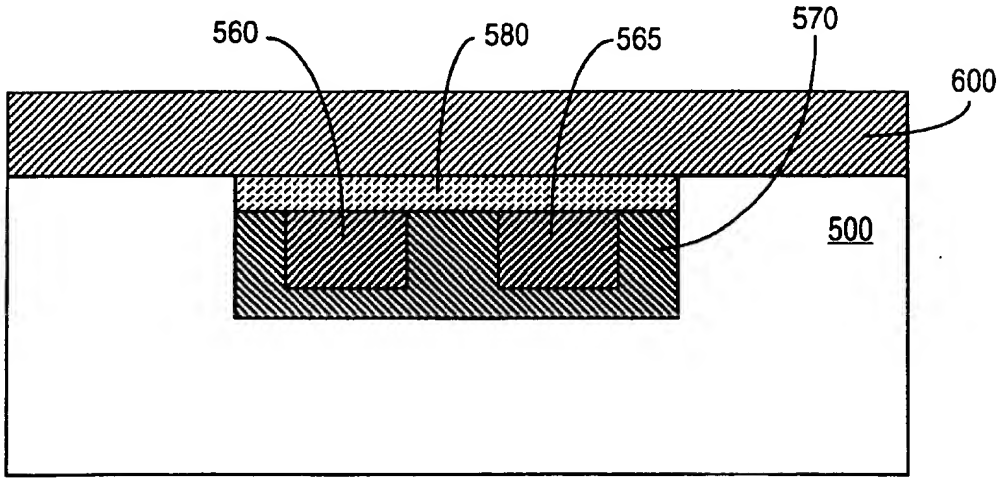

560
580
565
570
600
500

FIG. 16

METHOD FOR DISTRIBUTING A CLOCK ON THE SILICON BACKSIDE OF AN INTEGRATED CIRCUIT

This application is a division of Ser. No. 08/938,486 filed Sep. 30, 1997 now U.S. Pat. No. 6,037,822.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates generally to integrated circuit clocking and, more specifically to clock signal distribution in integrated circuits.

2. Description of Related Art

An issue facing the integrated circuit industry today is the problem of distributing clock signals throughout an integrated circuit die with low clock skew. Clock skew is the difference in arrival times of clock edges to different parts of the chip. Synchronous digital logic requires precise clocks for latching data. Ideal synchronous logic relies on clocks arriving simultaneously to all the circuits. Clock skew reduces the maximum frequency of the circuit as the circuit must be designed for the worst case skew to operate reliably.

The challenge facing integrated circuit designers is to insure that the clock switches at exactly the same time throughout the chip so that each circuit is kept in step to avoid delays that can cause chip failure. In prior art global clock distribution networks, clock skew caused by signal routing is typically controlled by the use of hierarchical H-trees. FIG. 1 is a diagram illustrating such a hierarchical H-tree clock distribution network **101** that is implemented in high-speed integrated circuits to reduce the clock skew effect. As shown in FIG. **1**, a clock driver **103** is used to drive H-tree network **101** at center node **105**. It is appreciated that clock driver **103** is typically a very large driver in order to provide sufficient drive to H-tree network **101**, which typically has a large capacitance in complex, high-speed integrated circuits as will be described below. As observed in FIG. **1**, the clock paths of the "H" formed between nodes **107**, **109**, **111**, and **113** have equal length between center node **105** and each of the peripheral points of the "H" at nodes **107**, **109**, **111**, and **113**, respectively. Therefore, assuming a uniform propagation delay of a clock signal per unit length of the H-tree network **101**, there should be no clock skew between the clock signal supplied to nodes **107**, **109**, **111**, and **113** from clock driver **103**.

FIG. 1 further illustrates H-tree network **101** taken to another hierarchical level with an "H" coupled to each respective peripheral node of the first level "H". Accordingly, every peripheral node **115** is an equal distance from node **107**. Every peripheral node **117** is an equal distance from node **109**. Every peripheral node **119** is an equal distance from node **111**. Finally, every peripheral node **121** is an equal distance from **113**. Thus, the clock paths from all nodes labeled **115**, **117**, **119**, and **121** are an equal distance from clock driver **103** and therefore should have no clock skew between them (assuming a uniform propagation delay) since the clock delay from clock driver **103** should be equal at all peripheral nodes of the H-tree network **101**. Thus, each node **115**, **117**, **119**, and **121** can be configured to act as a receiving station for a clock signal and service the clocking requirements of an area of the integrated circuit near the node with negligible clock skew with reference to other similarly configured nodes of the H-tree network.

As described, the H-tree propagation delay of a clock signal per unit length of the network may be controlled by placing every peripheral node an equal distance from clock driver **103**. However, the propagation delay of a clock signal because of length or distance of the paths traveled by the signal is only one ingredient that leads to skew. Another equally important ingredient is the consistency of speed of the signal as it traverses the path. One component that affects the speed of this signal is the resistance of the metal. Metal layers, such as Aluminum (Al), have an inherent resistivity that is a property of the metal, but the actual resistance a signal encounters can be affected by the thickness of the metal layer, because resistance is inversely proportional to layer thickness. In general, however, clock metal layer thickness is approximately 1.5 microns (μm) making the resistance of the metal fairly consistent or predictable in most integrated circuits.

The consistency of the speed of the signal in prior art clock distribution networks also depends generally on the impedance the signal encounters as it travels from the clock driver to the receiving station or clocked circuit. For a modern integrated circuit, there could be five or more metal interconnect layers on a chip, each interconnect layer separated from the other by a dielectric layer. The conventional clock network, such as H-tree network **101** overlays this structure. The clock network is laid out on a dielectric preferably overlying a ground plane metal. The speed of the signals along the path of the network is governed by the capacitance created in the dielectric between the clock network and the ground plane metal. Further, this capacitance is not consistent or uniform across the chip. This is so because the topography of a given chip gives rise to local variations, such as variations in the thickness of interlayer dielectric material relative to the underlying layer of metal and the presence of or absence of underlying metal layers. Interlayer dielectric thickness is important relative to the next level of metal. Further, the capacitive coupling from nearby switching lines adds to or subtracts from the clock signal development. The described variations inherent in a chip illustrate that the capacitance is dynamic, and therefore it is difficult to control the impedance encountered by the clock network, and thus the signal speed. In general, there is an inherent raw skew in the H-tree network due to this variation in signal speed of at least 150–200 pico seconds.

One effort to eliminate the skew caused by delays in signal speed is through the use of variable delay buffers (also referred to as deskew buffers) at the ends of the H-tree. The additional intentional skew introduced by these special buffers is controlled by a carefully distributed reference clock whose skew is made small. In this way, the main clocks at each of the endpoints of the H-tree are synchronized to the low skew reference clock. Although this scheme is very effective in reducing clock skew, the deskew buffers can cause additional jitter on the main clock due to the presence of any power supply noise internal to the chip. Hence, reduced skew is traded for increased jitter.

A second effort to eliminate or reduce timing skew is to use copper (Cu) as the interconnect metal forming the clock distribution. Since the consistency of the signal propagation is a function of the product of the capacitance and resistance, reducing the resistance reduces the sensitivity of the signal propagation to variations in the capacitance. The resistance of copper interconnect can be up to 50% lower than that of conventional Al-0.5% Cu. However, as the clock rate keeps climbing, even the resistance improvements provided by copper metallization may not be sufficient to control skew.

Even with sophisticated clock network configurations like the H-tree network, deskew circuits, and copper metallization, integrated circuits typically have a skew budget built into them that allows the circuits to tolerate a

certain amount of skew after which point the processing speed must be reduced. A general rule of thumb for a skew budget is a clock skew of 10% of the clock frequency.

In addition to clock skew and jitter, the clock distribution on the chip consumes valuable routing resources on integrated circuits that could be better used for signals and improved signal routability. As noted above, a preferred clock network routing is on top of a chip above a ground plane metal layer and separated by a dielectric layer. This preferred routing requires two layers of metal.

In addition to integrated circuit die area, the global clock distribution networks utilized today consume an increasing amount of power. If the total capacitance of the clock network is C, the power dissipated is CV^2f where V is the supply voltage and f is the frequency. The global clock distribution network on today's high-speed integrated circuit chips typically accounts for approximately 10% of the chip power.

The clock distribution network on a chip must be compatible with the chip package. The conventional packaging of a chip is illustrated in FIG. **2**. FIG. **2** is an illustration of a chip **205** packaged in a wire bond package **211**. As shown in FIG. **2**, wire bonds **203** for example, gold wire bonds, are used to connect package **211** and chip **205**.

Another type of packaging, is the Controlled Collapse Chip Connection (C4) packaged chips (sometimes referred to as flip-chip). FIG. **3** is an illustration of a C4 package **251**. C4 is the packaging of choice for high frequency chips as it provides high density, low inductance connections using ball bonds **253** between chip **255** and package **261** by eliminating the high inductance bond wires that are in wire bond packages.

SUMMARY OF THE INVENTION

A method and apparatus for clocking an integrated circuit is described. An apparatus includes an integrated circuit having a clock driver disposed in a first side of a semiconductor substrate, and a clock distribution network coupled to the clock driver and disposed in a second side of the semiconductor substrate to send a clock signal to clock an area of the integrated circuit. Additional features and benefits of the invention will become apparent from the detailed description, figures, and claims set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a schematic illustration of a prior art hierarchical H-tree clock distribution network that would be used to route clock signals above the chip circuitry.

FIG. **2** is a schematic illustration of a prior art wire-bond packaging technology.

FIG. **3** is a schematic illustration of a prior art flip-chip or C4 packaging technology.

FIG. **4** is a circuit diagram of a clock driver that is used in accordance with the invention to receive a signal from a master clock and transmit a single-ended clock signal to a clock network.

FIG. **5** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor substrate having a CMOS structure formed in the front side of the substrate and showing a masking layer overlying the backside of the substrate and a via formed in the substrate to diffusion/contacts for an embodiment of an integrated circuit in accordance with the invention.

FIG. **6** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor substrate having a CMOS structure formed in the frontside of the substrate and showing a dielectric material passivating the sidewalls of the via in the backside of the substrate in accordance with the invention.

FIG. **7** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor substrate having a CMOS circuit formed in the frontside of the substrate and showing a conductive material plug deposited in a passivated via formed in the backside of the substrate to diffusion, and a clock network line coupled to the plug in accordance with the invention.

FIG. **8** is a schematic illustration of a portion of an inverted semiconductor structure having a trench outlining a hierarchical H-tree clock network in the substrate backside for recessing a clock network.

FIG. **9** is a schematic illustration of a portion of an inverted semiconductor substrate in which a trench outlining a hierarchical H-tree clock network has been formed in the substrate the trench being passivated and filled with metal that forms the network in accordance with the invention.

FIG. **10** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor substrate having a CMOS circuit formed in the frontside of the substrate and showing a second dielectric layer overlying the clock network in accordance with the invention.

FIG. **11** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor substrate having a CMOS circuit on the frontside of the substrate and showing a heat sink coupled to the backside of the substrate in accordance with the invention.

FIG. **12** is a schematic illustration of a cross-sectional side view of a portion of an inverted substrate taken through line A—A of FIG. **11** in accordance with the invention.

FIG. **13** is a schematic illustration of a side view portion of a clock distribution network routed on the backside of a semiconductor substrate from a clock driver to a receiving station in accordance with the invention.

FIG. **14** is a schematic cross-sectional top view taken through line B—B of FIG. **13** and displaying a window through the heat sink and the network passivation layer.

FIG. **15** is a block diagram illustrating the distribution of a differential clock network on the backside of a semiconductor substrate.

FIG. **16** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor substrate showing a coplanar waveguide transmission line or waveguide to implement a differential clock network routed on the backside of the substrate.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus for clocking an integrated circuit in a semiconductor substrate is disclosed. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the invention. It will be apparent, however, to one having ordinary skill in the art that the specific detail need not be employed to practice the invention. In other instances, well known materials or methods have not been described in detail in order to avoid obscuring the invention.

The invention provides a method and apparatus for clocking an integrated circuit by routing the clock network along the backside of the semiconductor substrate and bringing clock signals through the backside of the semiconductor substrate to the diffusion/contact regions of the individual

devices that make up the integrated circuit and that are controlled by a master clock. The invention makes use of available area on the backside of a chip and is particularly compatible with flip-chip (C4) technology with the routing of the clock network on the substrate backside being completely compatible and functional with existing heat sink technology. The removal of clock network routing from the frontside of the integrated circuit frees up routing resources. Further, additional levels of metal are not needed to control clock skew, for instance, where an additional ground plane metal overlies the structure with the clock overlying the ground plane metal and separated by an interlayer dielectric.

By routing the clock network on the backside of the chip, clock skew can be minimized and controlled. The semiconductor substrate of the chip is a grounded conductor for which capacitance, and thus signal speed, can be completely controlled. Thus, the clock routing on the backside eliminates the need for deskew circuits and the accompanying jitter. By routing the clock over a dielectric having a low dielectric constant, the capacitance of the clock distribution may be reduced leading to reduced power or a smaller clock driver. Further, since the backside of the chip is a grounded plane, there is less impedance change.

FIGS. **4–12** illustrate one embodiment of a method of routing a clock network on the backside of a semiconductor substrate chip that is compatible with flip-chip (C4) technology.

FIG. **4** illustrates by way of a circuit diagram the distribution of a master clock signal through a clock network to multiple receiving stations. In this embodiment, the clock network is, for example, an H-tree that is routed on the backside of a chip. In this embodiment, the sequence of steps involved in fabricating such a structure are: (1) Generate, in the silicon substrate, vias that are filled with metal plugs and which form the medium for electrically connecting circuitry on the chip front side to the clock network on the chip backside; (2) fabricate trenches in the backside on the silicon substrate in the outline of the H-tree clock network; and (3) fill the trenches with metal that forms the clock network. These various steps will be described in detail below.

FIG. **4** includes a clock driver **375** that is an inverter circuit, such as a complimentary metal oxide semiconductor ("CMOS") field effect transistor. Clock driver **375** includes an NMOS device (with gate **310**, source region **320**, and drain region **325**) and a PMOS device (with gate **315**, source region **335**, and drain region **330**). NMOS gate **310** is connected to PMOS gate **315** and the two are connected to a master clock.

FIG. **4** shows a clock network in which a master clock drives clock driver **375**. Adjacent clock driver **375** are receiving inverters (labeled I, II, N) connected to clock driver **375** (illustrated by node **327**) to receive the clock signal.

FIG. **5** shows a schematic illustration of a cross-sectional side view portion of an integrated circuit structure. The integrated circuit structure has been inverted so that what is conventionally considered the frontside of the chip appears at the bottom of the figure. The portion of the integrated circuit illustrated in FIG. **5** shows a semiconductor substrate **300** of, for example, silicon, having embedded in substrate **300** and formed thereon a conventional complimentary metal oxide semiconductor ("CMOS") field effect transistor inverter. The CMOS inverter consists of both an NMOS device and a PMOS device, separated in this illustration by shallow-trench isolation techniques (denoted by dielectric-filled trenches labeled "STI"). It is to be appreciated that other isolation techniques, such as for example, Local Oxidation of Silicon (LOCOS), can be used to isolate the devices of the circuit.

In FIG. **5**, the gate **310** of the NMOS device is connected to the gate **315** of the PMOS device. NMOS gate **310** has adjacent source region **320** and drain region **325**. Similarly, PMOS gate **315** has adjacent source region **335** and drain region **330**. A polysilicon interconnect layer **332** lies adjacent to the PMOS and NMOS devices. A dielectric layer **337**, for example, a plasma deposited silicon oxide (SiO_2), overlies the structure. Vias are formed to the diffusion regions and gates. A conductive plug material **340**, such as for example, tungsten (W), fills the vias. Metal tracks **342**, in the first layer of metal interconnect using, for example, aluminum (Al/Cu), are coupled to plug material **340** to connect the adjacent gates **310** and **315**, respectively. FIG. **5** also illustrates metal tracks, e.g., Al/Cu tracks, **345** and **347** coupled to plug material **340** to the source of the NMOS and PMOS devices, respectively, to form contacts to these regions. Also, metal tracks **348**, e.g., Al/Cu tracks, are coupled to plug material **340** deposited in vias to drains **325** and **330** of the NMOS and PMOS devices, respectively, to connect drains **325** and **330** to one another. A second layer of dielectric **349**, for example, a silicon dioxide (SiO_2) glass, overlies the first metallization layer containing tracks **342**, **345**, **347**, and **348**, respectively. It is to be appreciated, that integrated circuits may have five or more metal interconnect layers. In FIGS. **5–12** only one metal layer, illustrated collectively as **342**, **345**, **347**, and **348**, respectively, is shown to facilitate understanding of the invention. The invention is suitable for devices having multiple levels of interconnect on the frontside of substrate **300**.

On the backside of substrate **300**, via **355** is formed to drain/contacts **325** of the NMOS device. The connection could also be made to the PMOS device or both the NMOS and PMOS devices. Connection to the NMOS device only is shown for clarity. First, a masking layer **350**, such as for example, a silicon nitride (Si_3N_4) masking layer **350**, is deposited over the backside of substrate **300** to protect substrate **300** from a subsequent etchant and to define a via. Next, the backside of substrate **300** is exposed to a suitable etchant to form via **355** in substrate **300** to drain **325**. Since the substrate thickness is likely. greater than 500 microns (μm), a fast etch method is used to etch via **355**. A suitable etch method is wet anisotropic etching of the silicon along the 111 planes using, for example, potassium hydroxide (KOH). Such an etch method generates a tapered hole. Plasma etching could also be utilized, such as for example, using an SF_6 etch chemistry in a reactive ion etcher (RIE) or an electron cyclotron resonance (ECR) etcher.

FIG. **5**, the CMOS circuit shown is a clock driver **375**. A clock signal is delivered, for example, to gate **342**. The output of clock driver **375**, i.e., drain **330** and drain **335** connected to metal **348**, is connected to the clock network.

Once via **355** is formed in the backside of substrate **300** to the inverter that is, for example, clock driver **375** or other clocked circuit, FIG. **6** illustrates the further processing steps to route the clock on the chip backside. FIG. **6** is a schematic illustration of a cross-sectional side view portion of inverted semiconductor substrate **300**. In FIG. **6**, masking layer **350** has been removed, and a dielectric interface layer **360** is formed along the side wall of via **355** to passivate via **355**. Dielectric interface layer **360** may be deposited by conventional techniques, such as for example, chemical vapor deposition of dielectric material, or may be grown, for example, such as, thermal SiO_2. Dielectric interface layer **360** seals off any exposed silicon in via **355** and passivates via **355**. Dielectric interface layer **360** serves as an interface between substrate **300** and the conducting material that will ultimately fill via **355**. At this point, any dielectric material formed in the bottom of the via is removed, for example, by an anisotropic etch, such as a reactive ion etch.

In one embodiment, a dielectric material **360** with a low dielectric constant, on the order of 4.1 or less, is formed

along the sidewalls of via **355**. The low dielectric constant material reduces the capacitive load required of the clock network so the power requirements of the clock driver can be reduced, since the power (the CV^2f power) dissipated is completely due to the capacitive load that is driven. Dielectric materials with low dielectric constants include, but are not limited to, SiO_2, SiO_2 xerogel, fluorinated amorphous carbon, fluorinated SiO_2, various fluorinated polymers, and hydrogen silisesquioxane (HSQ). A barrier layer, such as titanium nitride, as used in conventional CMOS processing, may be applied to the sidewalls to improve adhesion between a conductive material and the via sidewalls.

FIG. **6** also illustrates the subsequent processing step of depositing a conductive plug material **370**, such as for example tungsten (W), into via **355**. The substrate backside is then etched, for example, by a plasma etch, to remove excess plug material and interface layer material from the substrate backside.

Once the through-silicon vias are formed and filled with metal plugs, FIG. **7** illustrates the next step to fabricate the actual H-tree on the backside of the silicon substrate. In one embodiment, the invention contemplates that the clock network be recessed into the backside of substrate **300** so as to be compatible with existing heat sink technology. FIG. **8** shows a schematic top view illustration of a portion of a semiconductor substrate having a trench **358** outlining a portion of hierarchical H-tree clock network formed into the backside of semiconductor substrate **300**. In this example, via **355** represents a via to clock driver **375** which is the CMOS inverter illustrated in FIG. **5** (and the circuit diagram illustrated in FIG. **4**) filled with conductive plug material **370**. Since the via hole is tapered and not drawn to scale, it is to be appreciated that the diameter of the hole on the silicon backside can be much bigger than the width of the interconnect. The recessed network connects the clock driver to the receiving stations about the chip.

The recessed clock network illustrated in FIG. **8** may be formed by conventional trenching techniques. For example, the network may be formed by applying a masking layer, such as for example, silicon nitride (Si_3N_4), to the backside of substrate **300** and exposing an area on substrate **300** that will accommodate the clock network. Alternatively, a direct write, silicon micromachining technology such as chlorine-enhanced laser etching or laser ablation could also be used. A short recess, for example, a shallow trench equivalent to the desired thickness of the fill material, is etched into substrate **350**. Next, the sidewalls of the clock network trench are passivated and, as illustrated in FIG. **7**, a conductive metal line **376**, such as for example aluminum (Al/Cu), that is the clock network is deposited over the top of conductive material **370** to make an electrical connection to conductive material **370**.

FIG. **7** shows clock network **376** overlying conductive plug material **370** in via **355**. Clock network **376** is separated or isolated from substrate **300** by dielectric layer **360**. Clock network **376** is also recessed relative to the surface of substrate **300**. Recessing of clock network **376** is optional and will allow the subsequent attachment of a heat sink to the backside of substrate **300** with minimal disruption of existing processes.

As noted above with respect to FIG. **8** and the accompanying text, a shallow trench can be formed in substrate **300** to define clock network **376**, such as for example an H-tree network as illustrated in the schematic top view of FIG. **9**, to recess clock network **376** in substrate **300**. As noted with respect to FIG. **6** and the accompanying text, the trench is shallower than the vias formed to the individual receiving stations of the network. It is to be appreciated that the depth of the clock network outline trench is primarily a function of the desired thickness of the clock network conductive material **376**.

The invention contemplates that the metallization layer **376** that forms the clock network may be thicker than conventional metallization layers. A thicker metallization layer **376** (i.e., a larger aspect ratio), such as for example, on the order of 5.0 microns (as compared to 1.5 microns as in the prior art), reduces the resistance of the metal, because resistance is inversely proportional to metal layer thickness (R=resistivity/layer thickness). Lowering the resistance therefore reduces the "RC delay," which is a common measure of chip circuit speed.

FIG. **10** is a schematic illustration of a cross-sectional side view portion of the inverted semiconductor substrate. In FIG. **10**, dielectric layer **380** is formed over clock network **376**. Dielectric layer **380** may be deposited by conventional techniques, such as for example, chemical vapor deposition of dielectric material. Dielectric layer **380** serves to passivate and protect the otherwise exposed portion of clock network **376** (such as for example an H-tree network) thus isolating clock network **376** from the heat sink that will subsequently be applied to substrate **300**, grease, air, etc. Suitable dielectric material for dielectric layer **380** includes SiO_2. Dielectric layer **380** may also be the same material as interface layer **360**.

FIG. **11** is a schematic illustration of a cross-sectional side view portion of the inverted semiconductor substrate. In FIG. **11**, heat sink **400** is attached to substrate **300**. Because the clock network, represented by conductive layer **376**, is recessed, heat sink **400** conforms to the backside of substrate **300** as in prior structures. Thus, routing clock network **376** on the backside of substrate **300** provides minimal disruption to conventional attachment of the heat sink to the substrate backside in C4 packaging technology.

FIG. **12** is a schematic illustration of a cross-sectional side view portion of inverted semiconductor substrate **300** taken through line A—A of FIG. **11**. FIG. **12** shows conductive plug material **370** connected to diffusion/contact region **325**. Conductive plug material **370** is electrically isolated from substrate **300** by dielectric interface layer **360**. Clock network **375** overlies conductive plug material **370** and is likewise electrically isolated from substrate **300**. Finally, dielectric layer **380**, which can be the same material as interface layer **360**, overlies clock network **376** to protect clock network **376** from heat sink **400**, grease, air, etc.

FIG. **13** is a schematic illustration of a cross-sectional side view portion of inverted semiconductor substrate **300**. In FIG. **13**, additional circuitry is shown disposed in substrate **300**. Inverter circuit **375** is the clock driver which is coupled to a master clock. Also displayed is a second inverter or receiving circuit **475** adjacent to a diffusion region or receiving station **425** for a clock signal from clock driver **375**. Receiving station **325** is electrically coupled to gate **430** of receiving circuit **415** by metallization layer **435**, such as for example by aluminum (Al) coupled to tungsten (W) filled vias to diffusion/contact region **425** and gate **430**, respectively. Receiving station **425** receives a signal from clock driver **375** on clock network **376** and transmits that signal to transistor gate **430**. Thus, FIG. **13** shows a portion (collectively labeled **480**) of the clock network **376** wherein a receiving circuit **475** is coupled to clock driver **375**.

FIG. **14** is a schematic cross-sectional top view taken through line B—B of FIG. **13**. FIG. **14** shows heat sink **400** overlying substrate **300**, with a portion **480** displaying a window through heat sink **400** and dielectric layer **380**. Portion **480** shows clock network **376** coupled to clock driver **375** and receiving circuit **475**.

The above discussion focused on a clock network that constituted a single-ended connection between the transmitter and the receiver. The signal return path in this case is complex and through the ground return path of the chip. The clock driver and receiver could also be connected together

with a differential connection in which the signal return path is precisely defined.

FIG. **15** schematically illustrates by way of a block diagram a differential clock network technique that can also be employed on the backside of a semiconductor device. FIG. **15** shows two single connections between driver and receiver of a differential clock routed on the backside of a semiconductor substrate. FIG. **16** is a schematic illustration of a cross-sectional side view portion of an inverted semiconductor (e.g., silicon) substrate **500**. Conductive metal (e.g., aluminum) lines **560** and **565** make up the signal paths for a differential clock routing on the backside of substrate **500**. For example, conductive metal line **560** carries the clocking signal from the clock driver while conductive metal line **565** carries the inverse of the clocking signal. Each line of the differential connection is configured similarly to the way a single-ended connection is configured. This arrangement is sometimes also referred to as a coplanar transmission line or waveguide. The chief structural difference between the two configurations, is that for a differential connection, two single connections are required between driver and receiver. Conductive metal lines **560** and **565** are isolated from one another and from substrate **500** by a dielectric material **570**. In one embodiment, dielectric material **570** has a dielectric constant on the order of 4.1 or less. Conductive material lines **560** and **565** are recessed in a clock network trench such that an overlay of dielectric material **580** is deposited to passivate the lines and protect the lines from heat sink **600**, grease, air, etc. FIG. **14** shows a coplanar waveguide formed on the substrate backside.

The differential clock routing utilizes a coplanar waveguide which offers better clocking properties than a single-ended connection since the return path may be controlled more precisely.

The advantages of routing the clock network on the substrate backside are many. First, clock skew due to inner layer thickness variations and electrical activity on lower level metal is eliminated. Second, since there is a near perfect ground plane (the silicon substrate) with no topography over it, the impedance control is excellent. Third, the deskew circuits are eliminated, since propagation delay is uniform. This saves real estate and complexity and eliminates residual jitter. Fourth, since the clock distribution is moved to the silicon backside, more routing space is available for signals, leading to a potentially smaller die size. Fifth, since there are no other non-clock signals nearby or any other metal routing layers above the clock network, the metal thickness of the network can be increased to a larger aspect ratio to reduce resistance and increase circuit speed.

It is to be appreciated that in addition to minimizing clock skew, backside signal routing as previously described can be used to route other types of signals in a similar manner, particularly critical timing signals, inside an integrated circuit chip.

In the preceding detailed description the invention is described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A method comprising:

forming a via in a second side of a semiconductor substrate to a driver circuit on a first side of the semiconductor substrate;

routing a signal network on the second side of the substrate to a plurality of nodes on the first side of the substrate;

depositing a conductive material in the via; and

coupling the conductive material to the signal network.

2. The method of claim **1**, further comprising, prior to depositing the conductive material, introducing a layer of dielectric material along a sidewall of the via.

3. The method of claim **2**, wherein the plurality of nodes are coupled to a plurality of transistors that form an area of an integrated circuit,

wherein forming a via in the second surface comprises forming a via to each of the plurality of transistors, and

wherein depositing the conductive material in the via comprises depositing conductive material in each of the via to each of the plurality of transistors.

4. The method of claim **3**, further comprising:

forming a trench in the second side, the trench having a bottom and sidewalls; and

routing the signal network in the trench, wherein introducing a layer of dielectric material includes, prior to routing the signal network in the trench, introducing a dielectric material along the bottom and about the sidewalls of said trench to passivate said trench.

5. The method of claim **4**, wherein the dielectric material is a first dielectric material, the method further comprising:

depositing a second layer of dielectric material over the signal network.

6. The method of claim **5**, further comprising planarizing the second layer of dielectric material.

7. The method of claim **2**, wherein the dielectric material has a dielectric constant of 4.1 or less.

8. The method of claim **1**, wherein the signal network is comprised of a conductive material and is deposited such that the thickness of the conductive material is greater than 1.5 microns.

9. The method of claim **1**, wherein the signal network is a single-ended clock network.

10. The method of claim **1**, wherein the driver circuit comprises a first clock driver to send a clock signal and a second clock driver to send the inverse of the clock signal, and wherein the signal network is a first clock distribution network, the method comprising:

forming a via to the second clock driver,

routing a second clock distribution network on the second side of the substrate;

depositing a conductive material to the second via; and

coupling the conductive material to the second clock distribution network on the second side of the substrate.

11. The method of claim **1**, wherein routing the signal network comprises recessing the signal network in the second side of the substrate.

12. A method comprising:

routing a signal network on a first side of a semiconductor substrate;

coupling the signal network to a first device and to a second device on a second side of the semiconductor substrate opposite the first side; and

passing a signal between the first device and the second device via the signal network.

13. The method of claim **12**, wherein routing the signal network comprises recessing the signal network in the second side of the substrate.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,448,168 B1
DATED : September 10, 2002
INVENTOR(S) : Rao et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 3,
Line 32, after "that are", insert -- used --.

Column 8,
Line 2, delete "376", insert -- 375 --.

Signed and Sealed this

Fourth Day of February, 2003

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

US006246277B1

(12) **United States Patent**
Nitta et al.

(10) **Patent No.: US 6,246,277 B1**
(45) **Date of Patent: Jun. 12, 2001**

(54) **SEMICONDUCTOR INTEGRATED CIRCUIT DEVICE**

(75) Inventors: **Yusuke Nitta**, Kokubunji; **Toshihiro Hattori**, Kodaira, both of (JP)

(73) Assignee: **Hitachi, Ltd.**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/209,006**

(22) Filed: **Dec. 11, 1998**

(30) **Foreign Application Priority Data**

Dec. 26, 1997 (JP) 9-359275

(51) **Int. Cl.**[7] **H03K 19/096**
(52) **U.S. Cl.** **327/292**; 327/295
(58) **Field of Search** 327/292, 293, 327/294, 295, 564, 565; 326/93

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,420,696	*	12/1983	Gemma et al.	327/239
5,043,792	*	8/1991	Adachi	257/773
5,264,746	*	11/1993	Ohmae et al.	326/93
5,376,842	*	12/1994	Honoa et al.	327/293
5,430,397	*	7/1995	Itoh et al.	326/101
5,519,351	*	5/1996	Matsumoto	327/295
5,537,498	*	7/1996	Bausman et al.	327/293
5,691,662	*	11/1997	Soboleski et al.	327/292
5,923,188	*	7/1999	Kametani et al.	327/295
6,020,774	*	2/2000	Chiu et al.	327/295

FOREIGN PATENT DOCUMENTS

8-274260	10/1996	(JP) .
9-307069	11/1997	(JP) .

* cited by examiner

Primary Examiner—Tuan T. Lam
(74) *Attorney, Agent, or Firm*—Antonelli, Terry, Stout & Kraus, LLP

(57) **ABSTRACT**

A semiconductor integrated circuit device for minimizing clock skew over clock wiring shortened for reduced wiring delays. A plurality of stages of clock drivers are provided on clock wiring paths ranging from a clock generator to flip-flops. Clock lines connecting upper stage clock drivers are equalized in length in the form of a tree structure, and clock lines connecting lower stage clock drivers are made as short as possible.

3 Claims, 9 Drawing Sheets

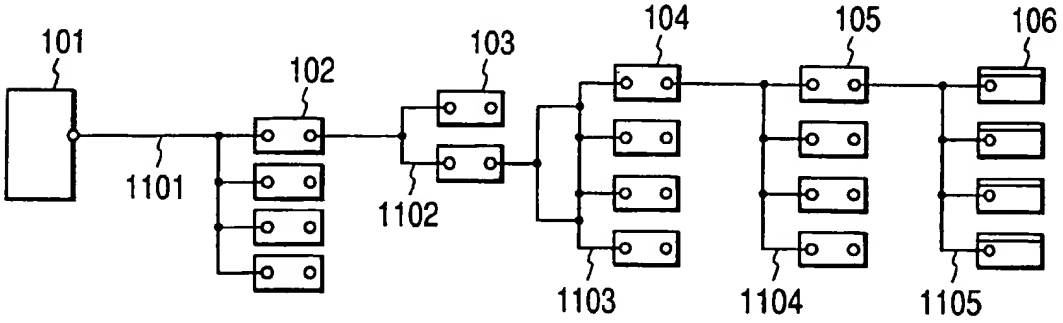


FIG. 1

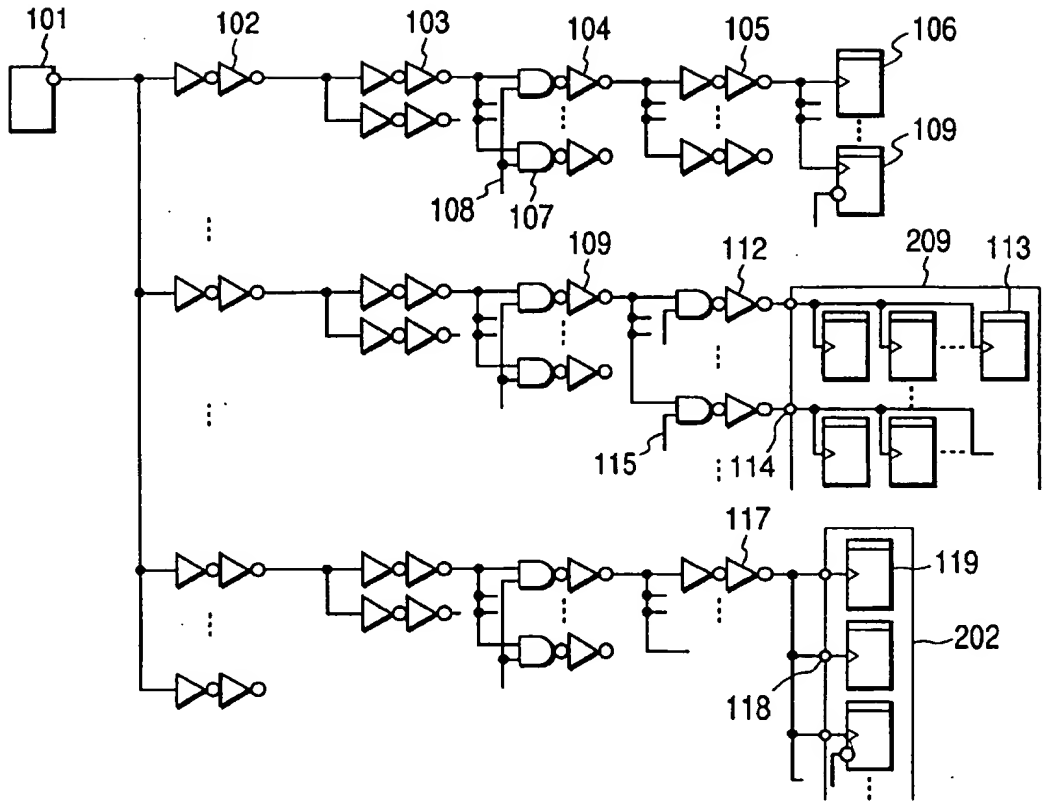

101
102
103
104
105
106
109
108
107
109
112
209
113
115
114
117
119
202
118

FIG. 2

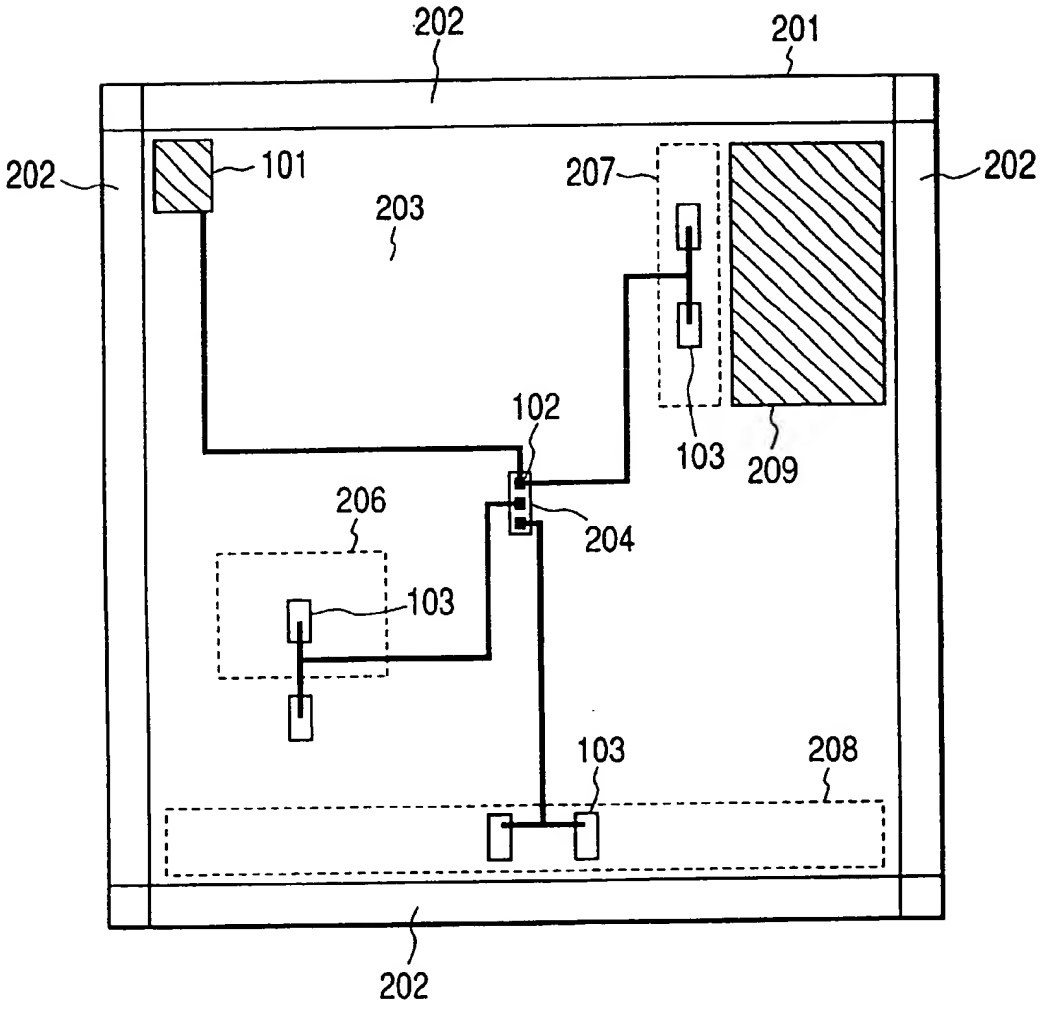


FIG. 3

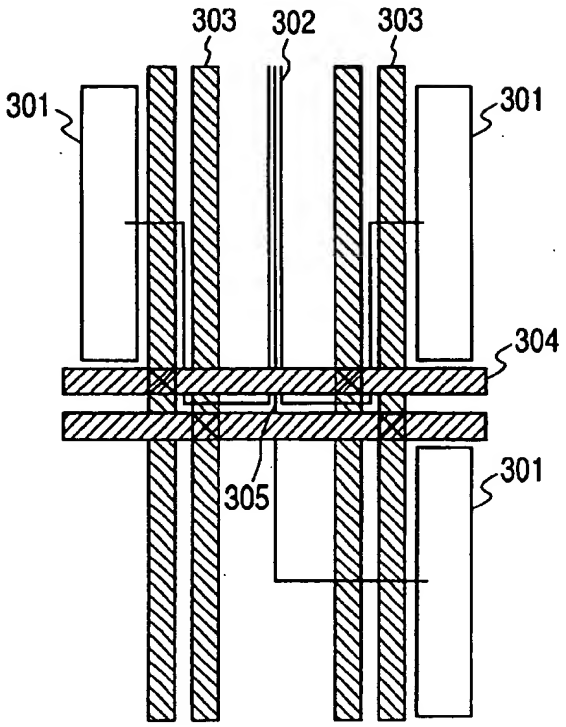


FIG. 4

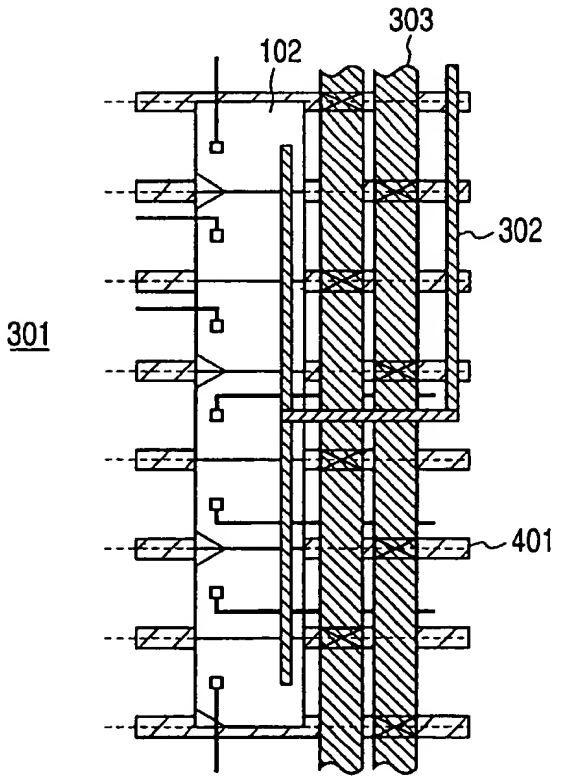


FIG. 5

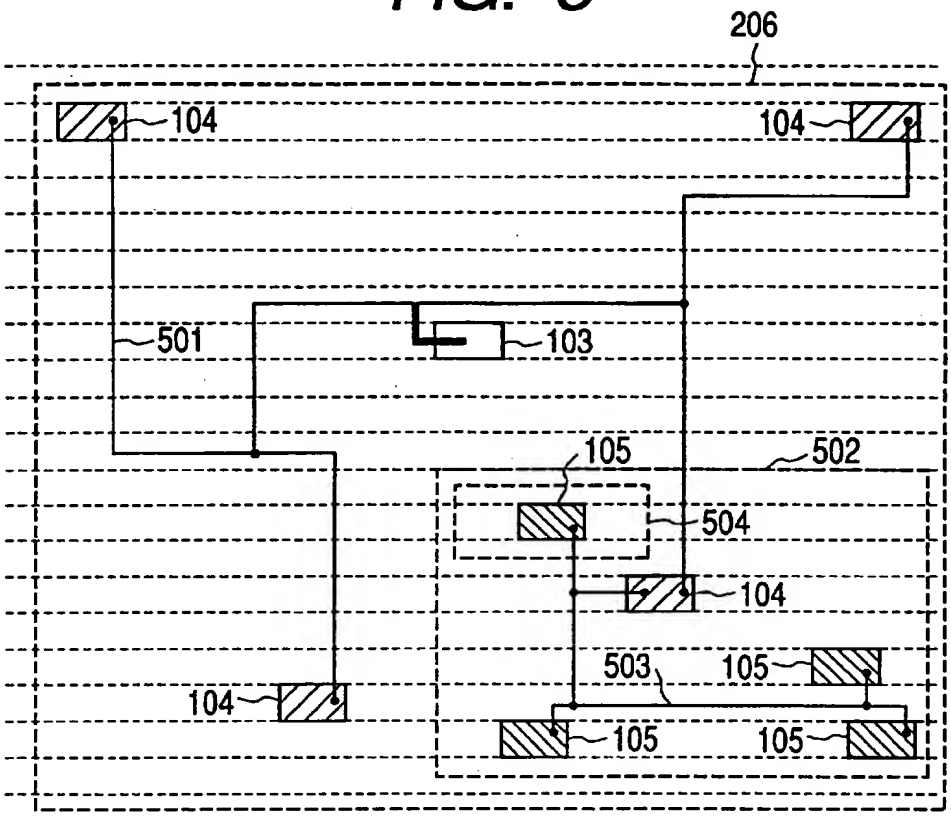

206
104
104
501
103
105
502
504
104
503
105
104
105
105

FIG. 6

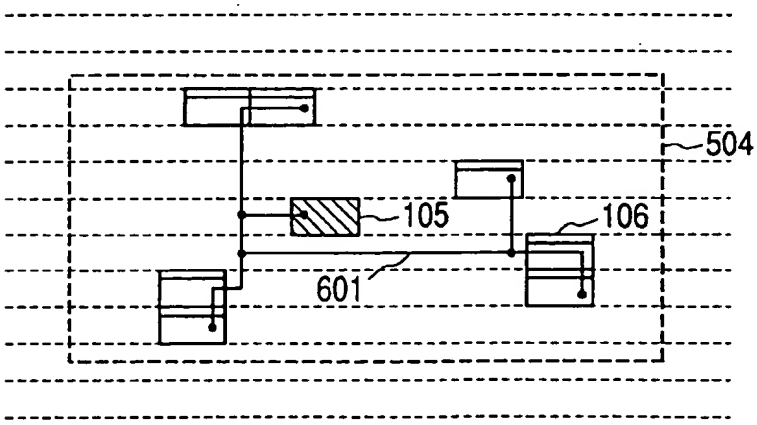

504
105
106
601

FIG. 7

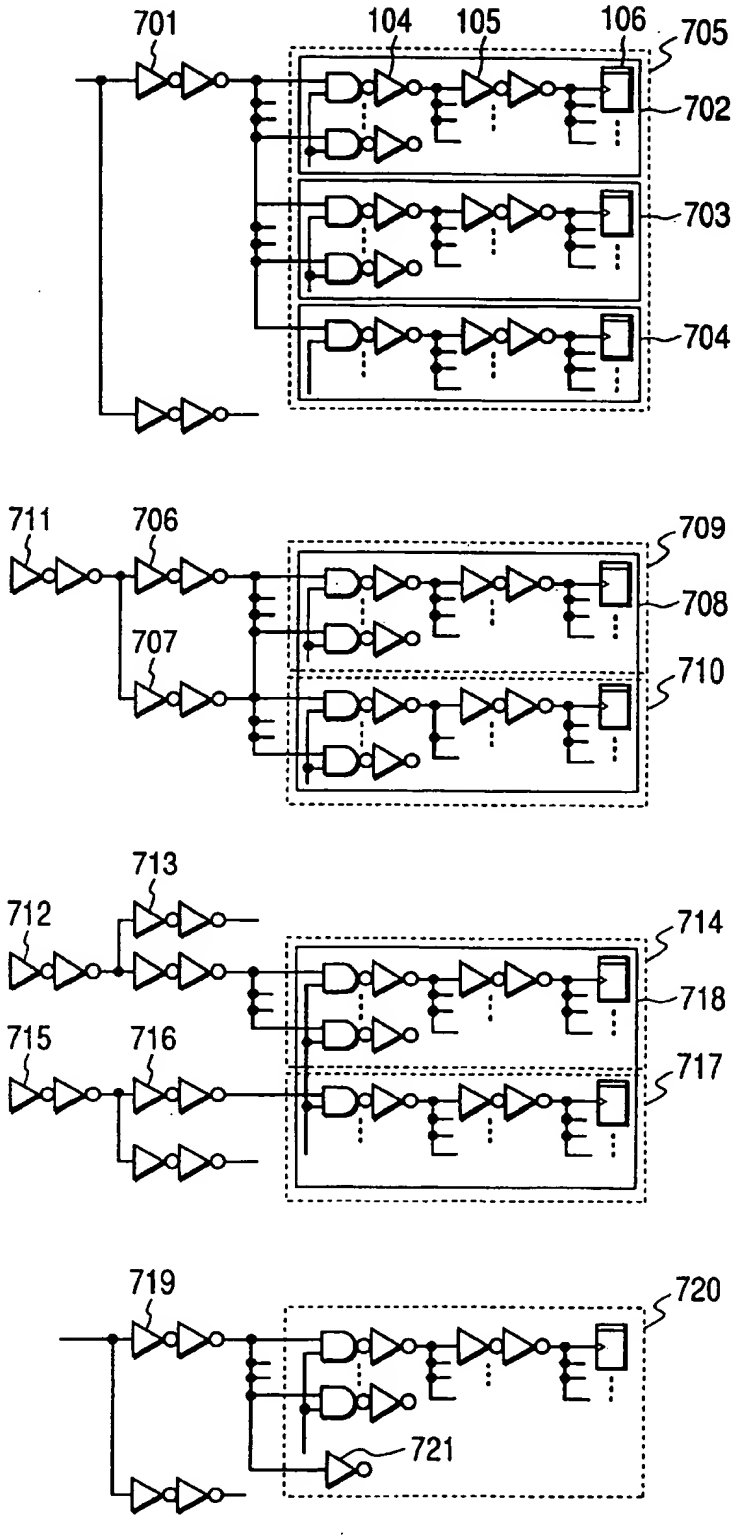

701
104
105
106
705
702
703
704
711
706
709
708
707
710
713
712
714
718
715
716
717
719
720
721

FIG. 8

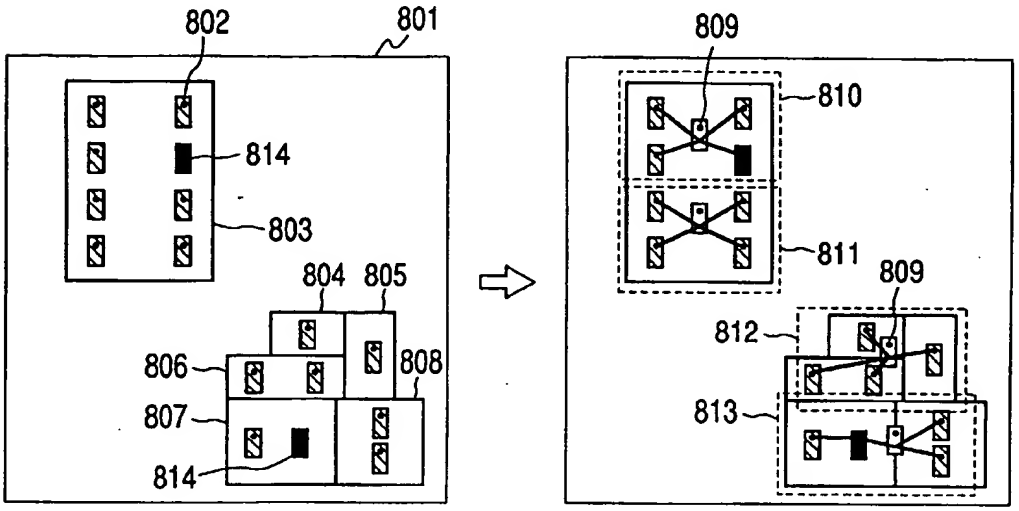

802
801
814
803
804
805
806
808
807
814
809
810
811
809
812
813

FIG. 9

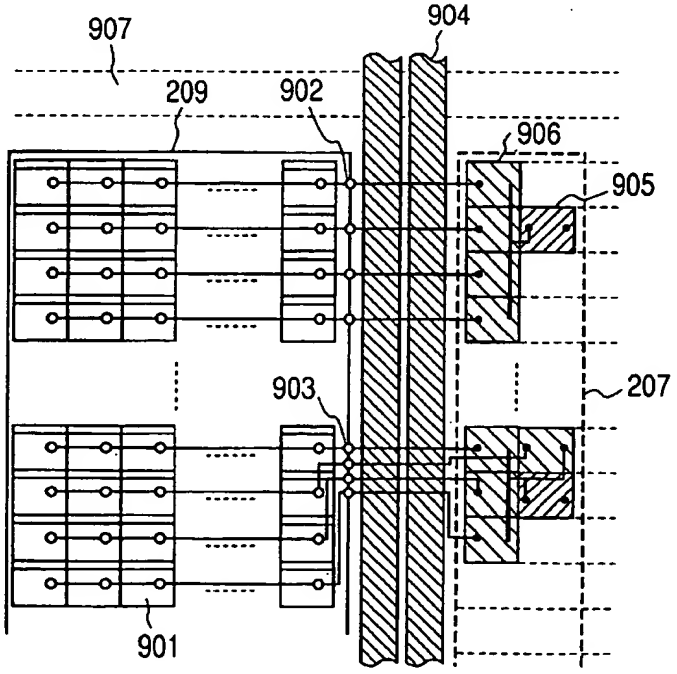

907
904
209
902
906
905
903
207
901

FIG. 10

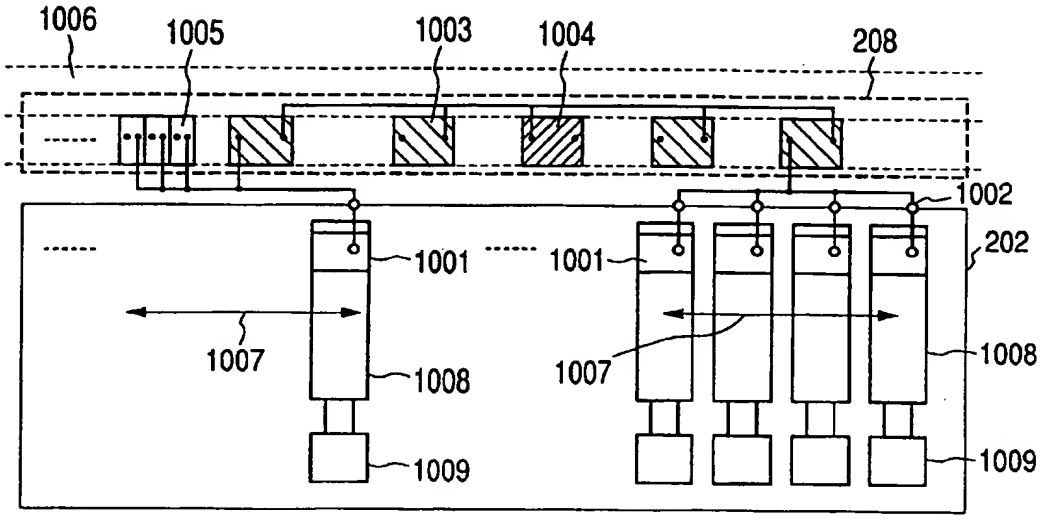

1006
1005
1003
1004
208
1002
202
1001
1001
1007
1007
1008
1008
1009
1009

FIG. 11

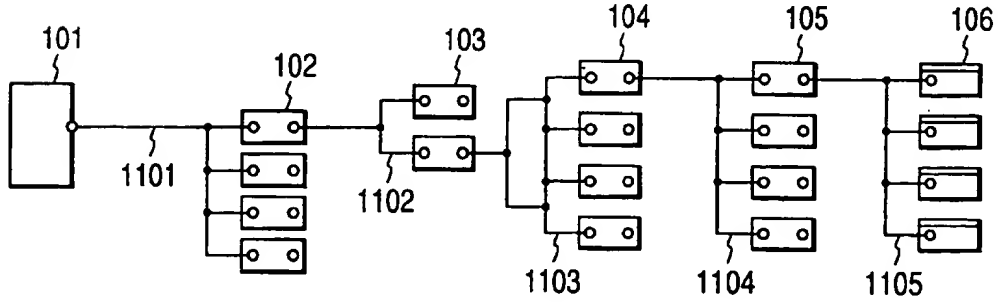

101
102
103
104
105
106
1101
1102
1103
1104
1105

FIG. 12

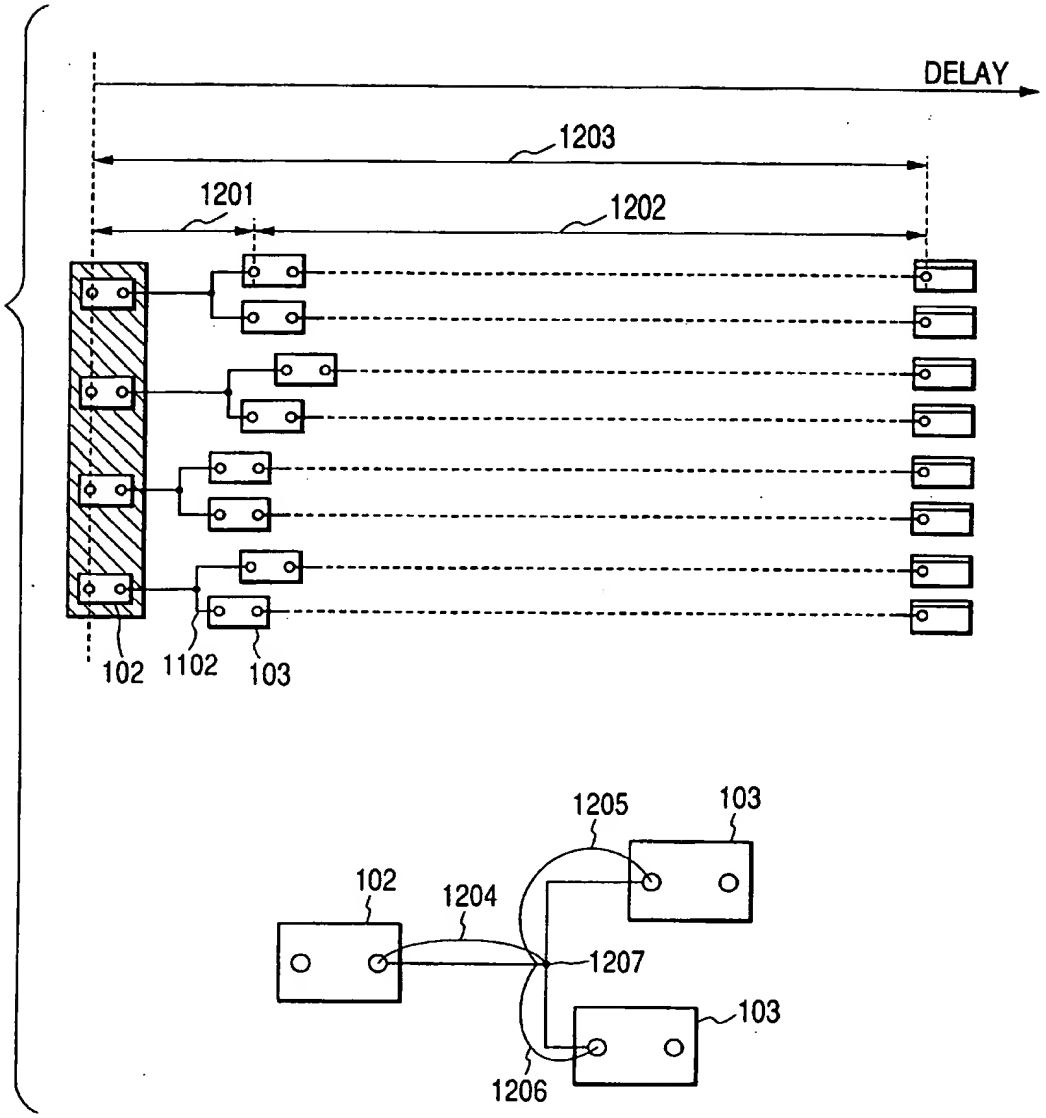

DELAY
1203
1201
1202
102
1102
103
1205
103
102
1204
1207
103
1206

FIG. 13

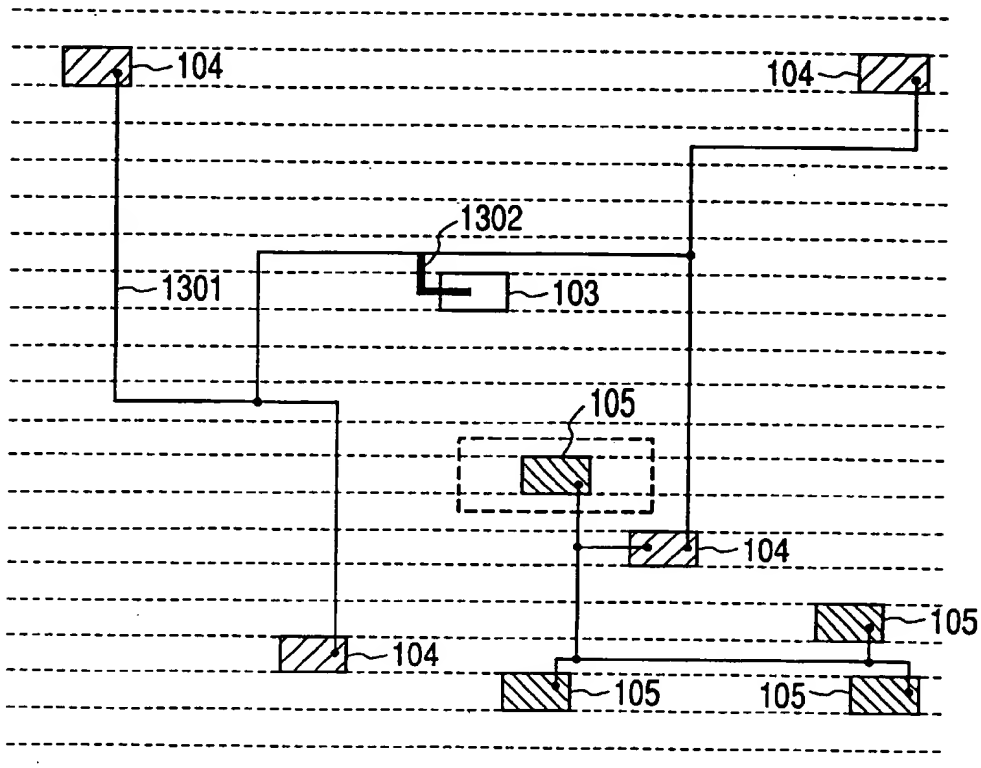
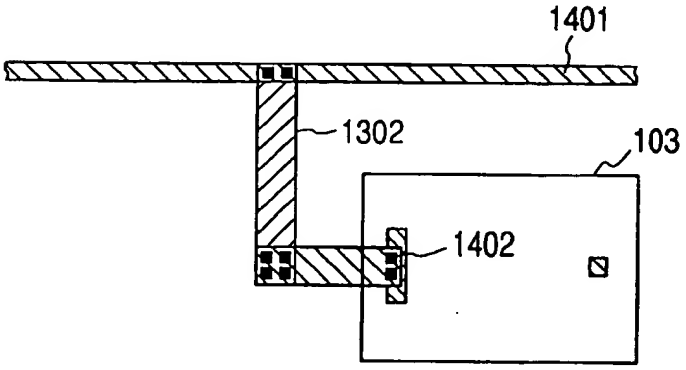

104
104
1302
1301
103
105
104
105
104
105
105

FIG. 14


1401
1302
103
1402

SEMICONDUCTOR INTEGRATED CIRCUIT DEVICE

BACKGROUND OF THE INVENTION

The present invention relates to a semiconductor integrated circuit device and, more particularly, to a semiconductor integrated circuit device having clock wiring with reduced clock skew.

Some semiconductor integrated circuit devices, such as VLSIs, include a synchronous circuit having flip-flops driven by a common clock signal. To make such a synchronous circuit operate more rapidly, these semiconductor integrated circuit devices require that clock skew (i.e., differences in clock supply timing between flip-flops) be minimized for removal of signal-to-signal timing differences.

Various layout design techniques for reducing such clock skew have been proposed. One such technique involves installing tree-structure paths between a clock signal generator and a plurality of flip-flops, wherein the length of the path between the generator and each flip-flop is suitably adjusted. Another technique, which is disclosed in Japanese Published Unexamined Patent Application No. Hei 9-307069, requires inserting clock buffers where appropriate when tree-structure wiring has been established, whereby the tree structure is readjusted so that the difference between a maximum and a minimum of delays on the readjusted wiring attains a predetermined value. Where there still remains clock skew despite the provision of tree structure wiring, another technique disclosed in Japanese Published Unexamined Patent Application No. Hei 8-274260 seeks to minimize the skew by replacing appropriate drivers with small-capacity drivers so that the paths with maximum skew become equal in skew level to other tree branch paths between second stage clock drivers and block circuits.

The conventional techniques outlined above have failed to consider optimum arrangements of skew reduction for VLSIs. These techniques presuppose that on tree-structure paths between a clock generator and each flip-flop, each node is afforded wiring of an equal length. If equal-length wiring is provided ranging from a clock generator through a plurality of stages of drivers to flip-flops, alternative lines necessitated by the equal-length lines at all stages prolong the overall clock wiring. The resulting disadvantages include more delays of clock signals and higher power dissipation.

Furthermore, the conventional techniques above have disregarded an optimum clock layout for each of the functional portions or for each of a plurality of clock phases in connection with LSIs. A VLSI comprises random logic circuits and data paths reflecting various functions of the device, as well as numerous I/O pads. The conventional techniques have so far shied away from providing any optimum clock layout for the diverse internal arrangements of the LSI.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a semiconductor integrated circuit device having a clock skew-lowering layout that ensures reduced wiring delays, enhanced packaging density and low clock power dissipation.

It is another object of the present invention to provide a semiconductor integrated circuit device having an optimum clock layout corresponding to each of the functional portions constituting an LSI.

These and other objects, features and advantages of the invention will become more apparent upon a reading of the following description and appended drawings.

Major features and benefits of the invention are outlined below. In carrying out the invention, and according to one aspect thereof, there is provided a semiconductor integrated circuit device comprising a plurality of stages of clock drivers furnished on clock wiring paths ranging from a clock generator to flip-flops. Clock lines connecting upper stage clock drivers have an equal length each in the form of a tree structure, and clock lines connecting lower stage clock drivers have the shortest possible lengths.

The lower the stage, the greater the number of clock drivers furnished. In that structure, clock lines connecting lower stage clock drivers are made to have not equal lengths but the shortest possible lengths. The arrangement shortens the overall clock wiring, reduces wiring delays, enhances packaging density, and lowers clock power dissipation. Since the lower stage clock drivers are connected by lines that are shorter than those connecting the upper stage clock drivers, the lower stage clock drivers may have the shortest possible wiring entailing negligible clock skew. Because the upper stage clock drivers are connected by extended wiring, the lines constituting such wiring are made to be equal in length in order to minimize clock skew.

A semiconductor integrated circuit device according to another aspect of the invention also comprises a plurality of stages of clock drivers. Of these drivers, intermediate stage clock drivers are provided with clock logic circuits for controlling clock signal supply.

The clock logic circuits control the supply of clock signals to individual function blocks corresponding to the intermediate clock drivers in question. The setup implements a clock signal supply scheme suitable for a VLSI while minimizing clock skew. Preferably, next-to-last stage clock drivers may have clock logic circuits for supply of clock signals to the flip-flops of random logic circuits and input/output pads, and both last stage and next-to-last stage clock drivers may have clock logic circuits for the supply of clock signals to the flip-flops of data paths.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a schematic circuit diagram of clock logic circuits applicable to a semiconductor integrated circuit device embodying the invention;

FIG. **2** is a top view of a clock layout on a chip carrying the semiconductor integrated circuit device embodying the invention;

FIG. **3** is a detailed plan view of the layout of a region **204** in FIG. **2**;

FIG. **4** is a more detailed plan view of the vicinity of a region **301** in FIG. **3**;

FIG. **5** is a detailed plan view of the layout of a region **206** in FIG. **2**;

FIG. **6** is a more detailed plan view of the layout of a region **504** in FIG. **5**;

FIG. **7** is a set of schematic views depicting relations between clock drivers at different stages on the one hand and logic blocks on the other hand in the inventive semiconductor integrated circuit device;

FIG. **8** is a set of layout diagrams illustrating relations between the regions handled by the second stage clock drivers shown in FIG. **7** on the one hand and logic blocks on the other hand;

FIG. **9** is a detailed plan view of clock drivers laid out in data paths;

FIG. **10** is a detailed plan view of clock drivers laid out in an input/output pad;

FIG. **11** is a conceptual diagram showing how different stages of the inventive semiconductor integrated circuit device are typically wired;

FIG. **12** is a set of explanatory diagrams indicating how differences between clock delays are reduced over different paths by use of clock wiring **1102**;

FIG. **13** is a schematic view depicting typical clock wiring ranging from second stage clock drivers to third stage clock drivers; and

FIG. **14** is a partially enlarged view of the clock wiring from the second stage clock drivers to the third stage clock drivers in FIG. **13**.

DETAILED DESCRIPTION OF THE EMBODIMENTS

FIG. **1** is a schematic diagram showing clock logic circuits applicable to a semiconductor integrated circuit device embodying the invention. This embodiment comprises four stages of clock drivers through which a clock signal generator **101** supplies clock signals to all flip-flops **106** inside the chip. The flip-flops are located in random logic circuits, data paths, and input/output pads.

The clock drivers at each stage play the roles described below. Clock drivers **102**, situated at the first stage as viewed from the clock signal generator, are called root clock drivers. These drivers distribute throughout the entire chip the clock signals output by the clock signal generator.

Second stage clock drivers **103** distribute clock signals to third stage clock drivers **104**. The drivers **104** are located in regions each made up of a number of logic blocks in the chip.

The third stage clock drivers **104** and fourth stage clock drivers **105** serve to distribute clock signals to all flip-flops in the logic blocks. If the third stage clock drivers **104** are constituted logically to control the supply of clock signals, it is possible to control clock signal supplies on a block-by-block basis.

Each third stage clock driver **104** supplies clock signals to a group of fourth stage clock drivers **105** distributed in each of the logic blocks. The fourth stage clock drivers **105** supply clock signals directly to the flip-flops **106**. Each driver **105** feeds clock signals to a group of flip-flops **106** distributed in the logic blocks of random logic circuits. Each data path **209** supplies clock signals to a column of flip-flops **113** via a clock terminal **114**. Each I/P pad portion **202** feeds clock signals to flip-flops **119** within a predetermined distance by means of clock terminals **118**.

The third stage clock drivers **104** are provided as AND circuits each having a control signal input terminal **107**. All third stage clock drivers **104** inside each of the logic blocks are connected to a signal line **108** that controls the supply of clock signals within the block in question. With the third stage clock drivers **104** provided as AND circuits, there is no need to provide each fourth stage clock driver **105** as an AND circuit. This minimizes the number of clock drivers that need to be replaced by AND circuits. A reduction in the number of clock drivers replaced by AND circuits directly translates into reduction in the lengths of the control signal lines.

The concept sketched in FIG. **1** is not limited to a single-phase clock scheme; it is obviously applicable to multi-phase clock arrangements as well. The semiconductor integrated circuit device of this embodiment includes three lines **302** coming from the clock signal generator and implementing a three-phase clock scheme, as shown in FIG. **3**. The three-phase clock scheme generates three kinds of clock signal: a first clock signal, the fastest clock signal fed to a CPU and an FPU in the semiconductor integrated circuit device; a second clock signal supplied to bus access controllers such as a DMAC (direct memory access controller) and I/O pads; and a third clock signal fed to peripheral controllers. In FIG. **3**, three root clock driver layout regions **301** are furnished to match the three clock signals of the three-phase clock scheme. For purpose of explanation, FIG. **1** indicates in unified fashion the three kinds of clock signal: one clock signal fed to the flip-flops **(106, 109)** of the random logic circuits; another clock signal supplied to the flip-flops **(113)** of the data paths; and another clock signal fed to the flip-flops **(119)** of the I/O pads. Of the three-phase clock signals, the first clock signal is sent to the random logic circuits and data paths, the second clock signal is given to the random logic circuits and I/O pads, and the third clock signal is delivered to the random logic circuits.

Some flip-flops admit control signals, while others do not. The random logic circuits and I/O pads contain both types of flip-flops. The data paths **209** have no flip-flops admitting control signals. Instead, all flip-flops arranged in each single column are controlled collectively by a clock driver **112** that serves as an AND circuit having a control signal input terminal **115**. Where control signals are input to the last stage clock drivers **112** on the data paths, the flip-flops inside the data paths have no need for control terminals. This structure enhances the packaging density of the embodiment.

A low clock skew state is brought about by unifying differences in arrival time between clock signals sent from the clock signal generator **101** to all flip-flops (called clock delays hereunder). To unify the clock delays requires adjusting both the driving force of clock drivers and the load capacities associated with the clock drivers. The load capacity of a clock driver is determined by the total sum of the capacity of a line connected to the driver in question and the capacity of the input terminal of a fan-out destination cell. In the logic setup of this embodiment, the driving forces of the clock drivers at each stage and the load capacities associated therewith are adjusted so as to unify the clock delays involved, thereby harmonizing all clock delays. The clock drivers **102** and **103** use cells of the same type throughout all paths, each driver having an identical fan-out count and an equal wiring length. The clock drivers **104** and **105** have different fan-out counts at each stage but share the same total capacity including wiring capacity, with the exception of the clock drivers **112** on the data paths. For example, if fan-out destination clock drivers are far away so that the wiring involved is necessarily long, the fan-out count tends to be small. Conversely, if clock drivers are nearby, the fan-out count is likely to be large. Each clock driver **112** on a data path has up to 32 flip-flops **113** within the path. Thus the clock drivers **112** have greater capacities than the clock drivers of the random logic circuits. For that reason, clock drivers with high driving forces are used at the last stage to harmonize the clock delays with those of the random logic circuits and I/O pads. Thanks to the above-described four-stage clock logic arrangement over all paths, clock delays may be adjusted at each stage.

Where multi-phase clock signals are used, similar logical structures are instituted. Because all phases are matched with like logical structures, there occurs little key skew between the multi-phase clock signals.

FIG. **2** shows a clock layout on a chip **201** carrying the semiconductor integrated circuit device embodying the

invention. In FIG. 2, the clock signal generator **101** is located in a corner of the chip **201** and adjacent to an I/O pad portion **202**.

All clock drivers are furnished in a cell layout region **203**. The root clock drivers **102** are gathered together in a region **204** near the chip center. The clock signal generator **101**, which is vulnerable to adverse effects from other circuits, is located peripherally in the chip. The root clock driver **102** located centrally in the chip extends clock wiring to the downstream root clock driver. This setup ensures stable supply of clock signals and makes it easier to minimize clock skew.

Some second stage clock drivers **103** are allocated to a region **206** that comprises a number of logic blocks. Second stage clock drivers **103** assigned to the data path **209** are located in a clock driver layout region **207** on the clock terminal side of the data paths. Likewise, second stage clock drivers **103** destined for the I/O pad portions **202** are furnished in a clock driver layout region **208** on the clock terminal side of each pad.

FIG. **3** depicts in detail the layout of the region **204** in FIG. **2**. As mentioned earlier, what FIG. **3** portrays is a three-phase clock layout. The root clock drivers **102** are gathered together in regions **301** that are each adjacent to a power supply line **303**. Reference numeral **302** denotes signal lines coming from the clock signal generator.

In this multi-phase setup, the root clock drivers **102** for each clock phase flank a vertical power supply line **303** and a horizontal power supply line **304**. The clock lines **302** leading to the clock driver layout regions **301** for all clock phases run in parallel up to a point **305** where the lines are branched, the point **305** being at an equal distance from all clock driver layout regions. Because the clock driver layout regions are not concentrated on a single power supply line, the supply of power is stabilized. The wiring arrangement above makes the line lengths substantially equal for all phases between the clock signal generator **101** and each of the root clock drivers **102**.

FIG. **4** provides a more detailed view of the vicinity of one region **301** in FIG. **3**. As illustrated, the root clock drivers **102** in the region **301** are arranged adjacent to one another in the vertical direction. There is no other cell interposed between each root clock driver **102** and the power supply line **303**. With the root clock drivers **102** gathered near the chip center, the lines ranging from the clock signal generator to all root clock drivers **102** are made equal in length. Because the maximum distance between the root clock drivers **102** and the second stage clock drivers **103** is reduced, clock delays are lowered correspondingly. Where the regions **301** are located adjacent to the power supply lines, it is possible to supply power in a stable manner to the regions where a plurality of root clock drivers **102** are gathered together.

FIG. **5** gives a detailed view of the layout of the region **206** in FIG. **2**. The second stage clock driver **103** in the region **206** is located near the center of gravity of a plurality of third stage clock drivers **104** distributed within the same region. Lines making up a network **501** ranging from the second stage clock driver **103** to the third stage clock drivers **104** are equalized in length.

Each third stage clock driver **104** is allocated to a region **502** wherein fourth stage clock drivers **105** are gathered adjacent to one another. Lines constituting a network **503** ranging from the third stage clock driver **104** to the fourth stage clock drivers **105** are equalized in length.

FIG. **6** gives a more detailed view of the layout of a region **504** in FIG. **5**. As illustrated, each fourth stage clock driver **105** is allocated to the region **504** wherein flip-flops **106** are gathered adjacent to one another. Lines making up a network **601** ranging from the fourth stage clock driver **105** to the flip-flops **106** are equalized in length.

FIG. **7** provides a set of schematic views depicting relations between clock drivers at different stages on the one hand and logic blocks on the other hand in the inventive semiconductor integrated circuit device. Each of the regions **206** comprises either a plurality of logic blocks or part of a logic block. Logic blocks wherein the number of third stage clock drivers **104** is smaller than a reference fan-out count of the second stage clock driver **103** are gathered together; logic blocks wherein the number of third stage clock drivers **104** is larger than the reference fan-out count are each divided into smaller regions.

Illustratively, logic blocks **702**, **703** and **704** wherein the number of third stage clock drivers **104** is smaller than the reference fan-out count are gathered together in a region **705** handled by a second stage clock driver **701**. On the other hand, a logic block **708** in which the number of third stage clock drivers **104** is larger than the reference fan-out count is divided into regions **709** and **710**. The region **709** is handled by a second stage clock driver **706**, and the region **710** is dealt with by a second stage clock driver **707**. Reference numeral **711** in this setup denotes a root clock driver.

However, it is not desirable to establish a logical structure such as one of a region **718** that is divided into regions **714** and **717**, the region **714** being handled by a second stage clock driver **713** connected to a root clock driver **712**, the region **717** being dealt with by a second stage clock driver **716** coupled to another root clock driver **715**. This type of logical structure will give rise to a possibility that a single logic block can be subject to adverse effects of the clock skew over relatively long wiring between the clock signal generator and the root clock drivers.

Where the number of clock drivers is smaller than the reference fan-out count inside a region **720** handled by a second stage clock driver **719**, dummy cells **721** are added to the region to compensate for the shortage of clock drivers. A dummy cell is a cell of which the input capacity is the same as that of a clock driver connected to the same network and which does not use output signals of the network.

As described, the fan-out count of the second clock driver **103** may be taken as the reference value with respect to which adjustments are made as needed. This makes it possible to harmonize on all paths the clock delays stemming from the second clock drivers **103**.

FIG. **8** provides a set of layout diagrams illustrating relations between the regions handled by the second stage clock drivers shown in FIG. **7** on the one hand and logic blocks on the other hand. Illustratively, if the reference fan-out count of a second stage clock driver **809** is 4, then a logic block **803**, in which the number of third stage clock drivers **802** is greater than the reference fan-out count, is divided into regions **810** and **811**. Third stage clock drivers in each of the regions **810** and **811** are assigned a second stage clock driver **809**. How to divide a logic block is determined by the arrangement of third stage clock drivers **802** furnished therein. If the clock driver count in a divided region is smaller than the reference fan-out count, then previously furnished dummy cells **814** are used to take the place of third stage clock drivers **802** to compensate for the shortage of clock drivers. Meanwhile, each of logic blocks **804**, **805**, **806**, **807** and **808** has a smaller number of third stage clock drivers **802** than the reference fan-out count. In

such cases, adjacent logic blocks are gathered together to form a single region to which a second stage clock driver **809** is allocated.

In FIG. **8**, the logic blocks **804**, **805** and **806** are combined into a region **812**, and the logic blocks **807** and **808** into a region **813**. Where the number of third stage clock drivers **802** is smaller than the reference fan-out count inside a combined region, previously furnished dummy cells **814** are utilized to compensate for the shortage with respect to the fan-out count of the second stage clock driver **809**.

FIG. **9** is a detailed view of clock drivers laid out in data paths **209** shown in FIG. **2**. A clock terminal **902** is allocated to each column of flip-flops **901** in the data paths **209**. The clock terminals **902** are arranged so as to line up on one side of the data paths **209**.

A clock driver layout region **207** is provided on a cell layout region **907** on the side of the clock terminals **902** for the data paths. Inside the clock driver layout region **207** are third stage clock drivers **905** and fourth stage clock drivers **906**.

The clock driver layout region **207** is also arranged to be adjacent to a power supply line **904**. If clock drivers of the data paths are located on the cell layout region, it is possible to gather clock drivers together where the clock terminals are concentrated. This helps prevent a surge in clock delays. Providing the clock driver layout region forestalls increases of distances up to the clock drivers. Although the clock drivers of the data paths are considerably concentrated in terms of layout because of their numerous clock terminals, locating the clock driver layout region adjacent to the power supply line ensures stable supply of power.

Although not shown, there exist a large number of third stage clock drivers **905** of the data paths. In this setup, the wiring between the second stage clock drivers **103** and the third stage clock drivers **905** of the data paths is furnished as follows: a plurality of third stage clock drivers are grouped together, and the wiring within that group is made as short as possible. Lines between the second stage clock drivers **103** and the respective groups of third stage clock drivers are equalized in length.

FIG. **10** provides a detailed view of clock drivers laid out in the I/O pad portion **202** shown in FIG. **2**. A clock terminal **1002** is allocated to each flip-flop **1001** inside the I/O pad portion **202**. The clock terminals **1002** are arranged so as to line up on one side of the I/O pad portion **202**. A clock driver layout region **208** is furnished on a cell layout region **1006** on the side of the clock terminals **1002** in the I/O pad portion **202**. Inside the clock driver layout region **208** are third and fourth stage clock drivers **1004** and **1003** arranged in a row, each third stage clock driver being flanked by a plurality of fourth stage clock drivers. A reference wiring length is set for the fourth stage clock drivers **1003**, and as many clock terminals **1002** as a reference fan-out count are allocated within the reference wiring length. This arrangement is adopted here because the number of clock terminals are small despite the long distance occupied by them in the I/O pad portion **202**.

If there are fewer clock terminals within the reference wiring length **1007** than the reference fan-out count, then dummy cells **1005** are added to compensate for the shortage.

When the layout regions are furnished as described, any increases in the distances up to the clock drivers are substantially prevented. The use of numerous dummy cells makes it possible to harmonize clock delays despite the presence of sparsely arranged clock terminals.

The dummy cells **1005** should preferably be arranged in the same row as that of a plurality of clock drivers as illustrated in FIG. **10**. That is because the arrangement facilitates adjustment of the wiring lengths while minimizing increases in occupied areas.

In FIG. **10**, one fourth stage clock driver **1003** is furnished corresponding to four clock terminals. However, the one-to-four correspondence is not limitative of the invention. For example, suppose that each fourth stage clock driver **1003** is assigned 12 terminals and that only one flip-flop **1001** is connected to a fourth stage clock driver **1003**. In that case, 11 dummy cells **1005** may be connected to the fourth stage clock driver **1003** in question.

In the I/O pad portion **202**, each flip-flop **1001** is associated with an input/output circuit **1008** and an I/O pad **1009** which are arranged in the direction of a chip edge. In the inventive semiconductor integrated circuit device, the logic circuits inside of the I/O pad portions **202** use signals with an amplitude of 1.8 V, and are interfaced to signals with an amplitude of 3.3 V from outside the chip. The interface capability is implemented by use of a level shifter circuit arrangement. More specifically, each I/O circuit **1008** includes a three-state logic circuit, a level shifter circuit and an I/O buffer circuit arranged in that order starting from the flip-flop side. These circuits are connected to an I/O pad **1009**.

FIG. **11** gives a conceptual view illustrating how different stages of the inventive semiconductor integrated circuit device are typically wired. Lines **1102** and **1103** are equalized in length and constitute a binary tree structure. Lines **1104** and **1105** are made as short as possible. That is, the lines at a higher stage where fan-out destination cells are distributed extensively are equalized in length; wiring at a lower stage where fan-out destination cells are narrowly distributed is made the shortest possible wiring. Length differences (i.e., between a maximum and a minimum length) between clock lines equalized in length are smaller than length differences between clock lines that are made as short as possible.

Wiring **1101** is provided at the highest stage. However, since this wiring involves root clock drivers **102** gathered together as shown in FIG. **4**, it is prepared as the shortest possible wiring.

The lines **1102** and **1103**, with their fan-out destination cells distributed extensively, are equalized in length on all paths. This arrangement helps harmonize clock delays over the paths.

The above adjustments are made possible because the number of clock drivers the upper stages is limited. The lower the stage, the greater the number of clock drivers installed. Thus the lines are made as short as possible at lower stages in order to reduce the overall wiring length, boost packaging density and minimize line-induced delays. Because the wiring is shorter at lower stages, clock skew stemming from the line-induced delays is negligible there.

At higher stages where extended wiring promotes vulnerability to delays, the lines involved are equalized in length so as to reduce the clock skew caused by the line-induced delays. When all lines are equalized in length on all paths, differences in load capacity between clock drivers are eliminated.

FIG. **12** offers a set of explanatory views indicating how differences between clock delays are reduced over different paths by use of the clock wiring **1102**. It may happen that differences in clock delay **1202** exist between second stage clock drivers **103** and flip-flops **106**. Such differences, if they occur, are reduced by modifying the configuration of the lines **1102** which are basically equalized in length and which

constitute a binary tree structure. Specifically, clock delays **1201** are adjusted between the root clock drivers **102** and the second stage clock drivers **103**. For example, if there are clock delay differences between each of two second stage clock drivers **103** connected to a line **1102** on the one hand and the corresponding flip-flops **106** on the other hand, the lengths of lines **1205** and **1206** between a junction **1207** and the second stage clock drivers **103** are adjusted at the point **1207** in such a manner that the clock delay differences are removed. Any clock delay differences that may occur between another root clock driver **102** and the second stage clock drivers **103** are eliminated by adjusting the length of a line **1204** between the clock driver **102** and the junction **1207**. Such adjustments, which are relatively simple in procedure and required at only a small number of locations, may be carried out manually.

Where clock drivers of high driving forces are used, lines wider than usual need to be employed to counter migration. Because the incidence of migration is proportional to the strength of current, the wiring need only be composed of wide lines up to first junctions beyond which the current strength is reduced by half. Along clock wiring **1301** between a second stage clock driver **103** and third stage clock drivers **104** in FIG. **13**, a portion **1302** is made of a wide line (having twice the width of ordinary wiring) as shown in FIG. **14**. The rest of the wiring has the ordinary width such as that of a portion **1401**. An output terminal **1402** of each second stage clock driver **103** is shaped as a rectangle at least as broad as the wide line so that the latter may be connected properly to the terminal **1402**. Where there are a limited number of locations requiring wide-line wiring, packaging density is improved.

Wide-line wiring is not limited to the clock wiring between the second stage clock drivers **103** and the third stage clock drivers **104**. It is also possible to install wide lines up to the first junctions along the clock wiring between the root clock drivers **102** on the one hand and the second clock drivers **103** on the other hand.

As described, a semiconductor integrated circuit device having the inventive clock layout is subject to significantly reduced wiring delays, has increased packaging density, and provides a clock skew-lowering layout involving decreased clock power dissipation. The device also has an optimally arranged clock layout for each functional portion of the LSI.

As many apparently different embodiments of this invention may be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.

What is claimed is:

1. A semiconductor integrated circuit device comprising:

a clock signal generator for outputting clock signals;

a plurality of flip-flops for receiving said clock signals from said clock signal generator through clock lines; and

a plurality of stages of clock drivers furnished on clock lines ranging from said clock signal generator to said flip-flops;

wherein differences between a maximum and a minimum length of the clock lines between first stage clock drivers and second stage clock drivers are smaller than differences between a maximum and a minimum length of the clock lines between last stage clock drivers and said flip-flops.

2. A semiconductor integrated circuit device according to claim **1**, wherein the clock lines between said first stage clock drivers and said second stage clock drivers are equalized in length, and wherein clock lines between said last stage clock drivers and said flip-flops have the shortest possible lengths.

3. A semiconductor integrated circuit device comprising:

a clock signal generator for outputting clock signals; and

a plurality of stages of clock drivers furnished on clock lines coming from said clock signal generator;

wherein, at least either between first stage clock drivers and second stage clock drivers, or between second stage clock drivers and third stage clock drivers, clock lines up to first junctions are each made greater in line width than the corresponding clock lines beyond said first junctions.

* * * * *

US006367060B1

(12) **United States Patent**
Cheng et al.

(10) **Patent No.: US 6,367,060 B1**
(45) **Date of Patent: Apr. 2, 2002**

(54) **METHOD AND APPARATUS FOR CLOCK TREE SOLUTION SYNTHESIS BASED ON DESIGN CONSTRAINTS**

(76) Inventors: **C. K. Cheng**, 4407 Mensha Pl., San Diego, CA (US) 92130; **Liang-Jih Chao**, 40 Tissiack Ct., Fremont, CA (US) 94539

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/336,257**

(22) Filed: **Jun. 18, 1999**

(51) **Int. Cl.**[7] **G06F 17/50**
(52) **U.S. Cl.** **716/10**; 716/6
(58) **Field of Search** 716/1, 2, 4, 6, 716/7, 10; 703/19

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,140,526	A *	8/1992	McDermith et al.	364/488
5,410,491	A *	4/1995	Minami	364/491
5,519,351	A *	5/1996	Matsumoto	327/295
5,553,002	A *	9/1996	Dangelo et al.	364/489
5,557,779	A *	9/1996	Minami	395/500
5,740,067	A *	4/1998	Hathaway	364/489
5,963,728	A *	10/1999	Hathaway et al.	716/3

OTHER PUBLICATIONS

Minami et al., "Clock Tree Synthesis Based on RC Delay Balancing," IEEE 1992 Custom Integrated Circuits Conference, pp. 28.3.1–28.3.4.*

Balboni et al., "Clock Skew Reduction in ASIC Logic Design: A Methodology for Clock Tree Management," IEEE Trans. on CAD of ICs and Systems, vol. 17, No. 4, Apr. 1998, pp. 344–356.*

Cho et al., "A Buffer Distribution Algorithm for High–Performance Clock Net Optimizatin," IEEE Trans. on VLSI Systems, vol. 3, No. 1, Mar. 1995, pp. 84–98.*

Krishnamurthy et al., "A New Partitioning Framework for Uniform Clock Distribution During High–Level Synthesis," 1988 IEEE, pp. 381–384.*

Mehta et al., "Clustering and Load Balancing for Buffered Clock Tree Synthesis," 1997 IEEE, pp. 217–223.*

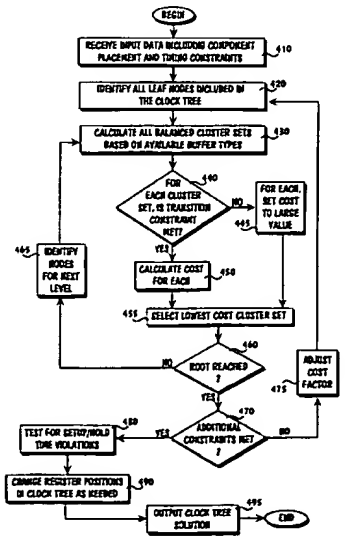
* cited by examiner

Primary Examiner—Matthew Smith
Assistant Examiner—Leigh Marie Garbowski
(74) *Attorney, Agent, or Firm*—Columbia IP Law Group, PC

(57) **ABSTRACT**

A clock tree synthesizer calculates balanced cluster sets of nodes a particular level of a clock tree in a circuit description based on a set of available buffer types. Each balanced cluster set is tested to see if it meets a design constraint. If the design constraint is not met for a particular balanced cluster set, the particular cluster set is removed from consideration in the clock tree solution. For the cluster sets that do meet the design constraint, a cost associated with each cluster set is calculated. A balanced cluster set that has the lowest cost is selected for the clock tree solution. In one embodiment, the lowest cost balanced cluster set for one level in the clock tree forms the nodes for the next higher level in the clock tree, and the process is repeated at each level of the clock tree up to a root node. In another embodiment, the clock tree in the circuit description is modified with the lowest cost balanced cluster set for each level of the clock tree solution, wherein each cluster includes the buffer on which the cluster calculation was based.

20 Claims, 8 Drawing Sheets



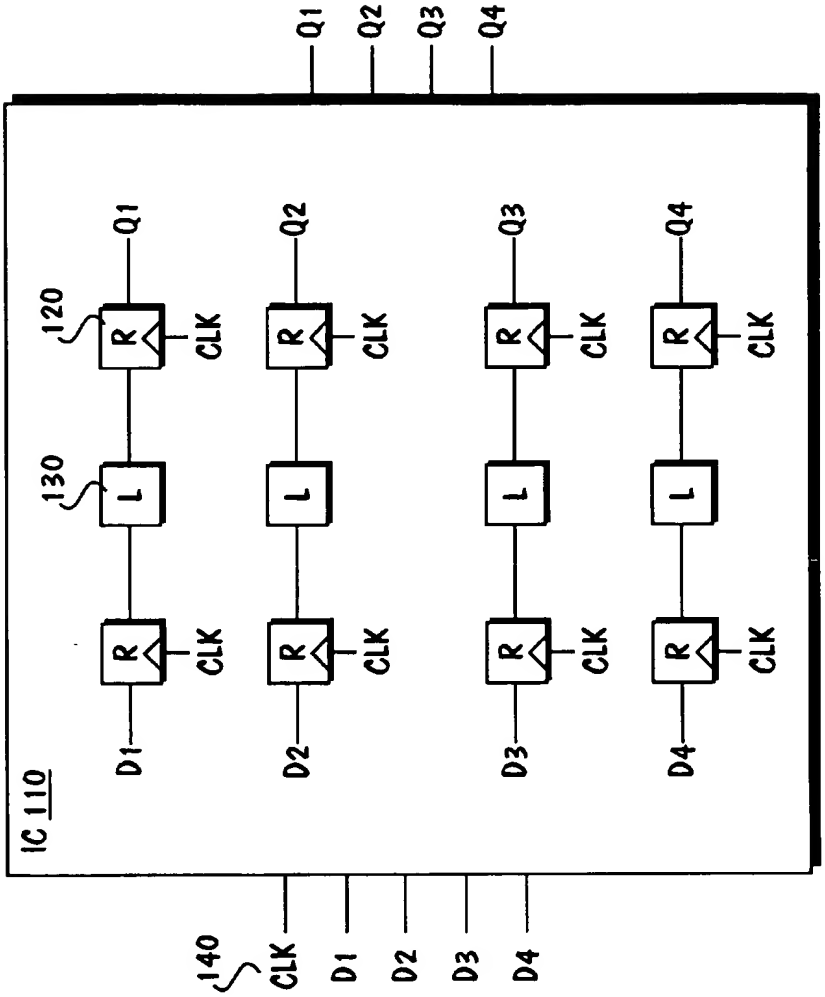


FIG. 1

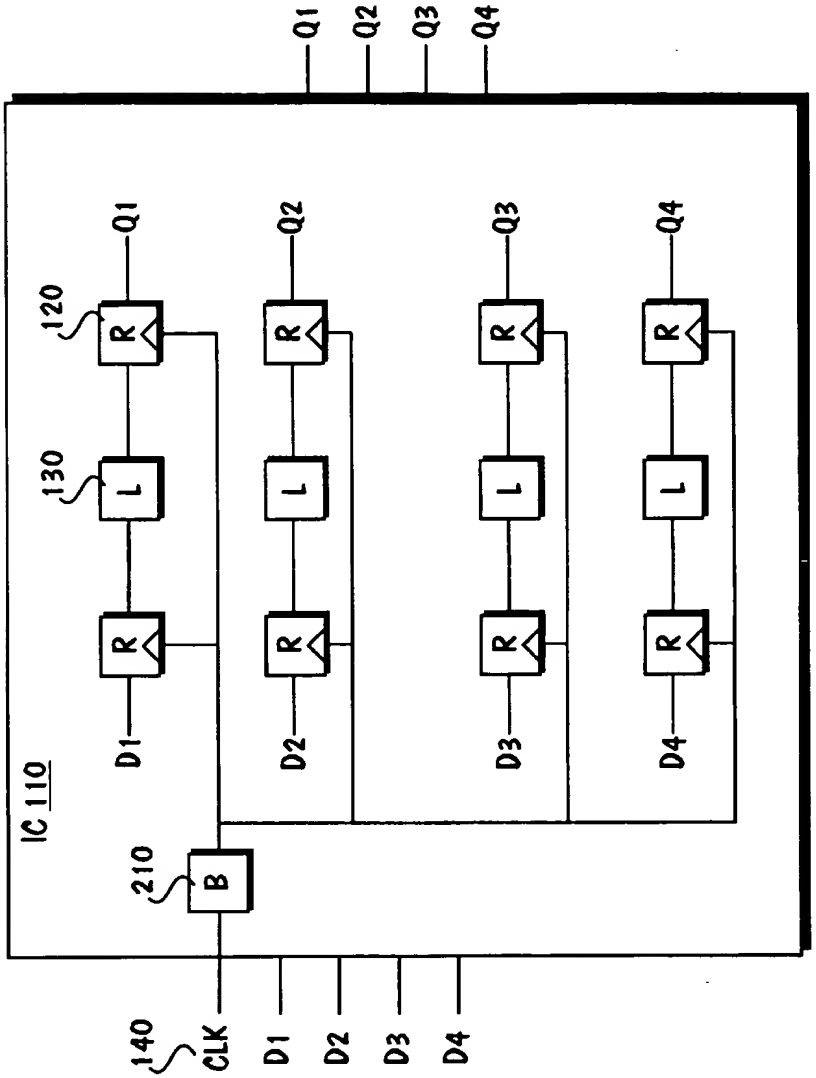


FIG. 2

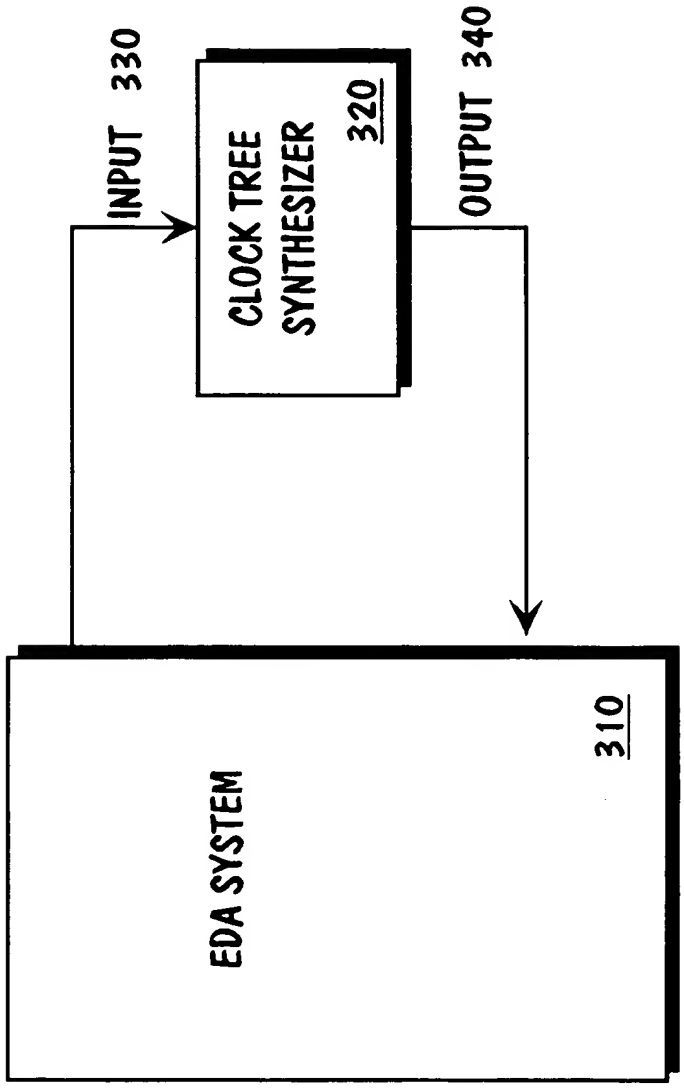
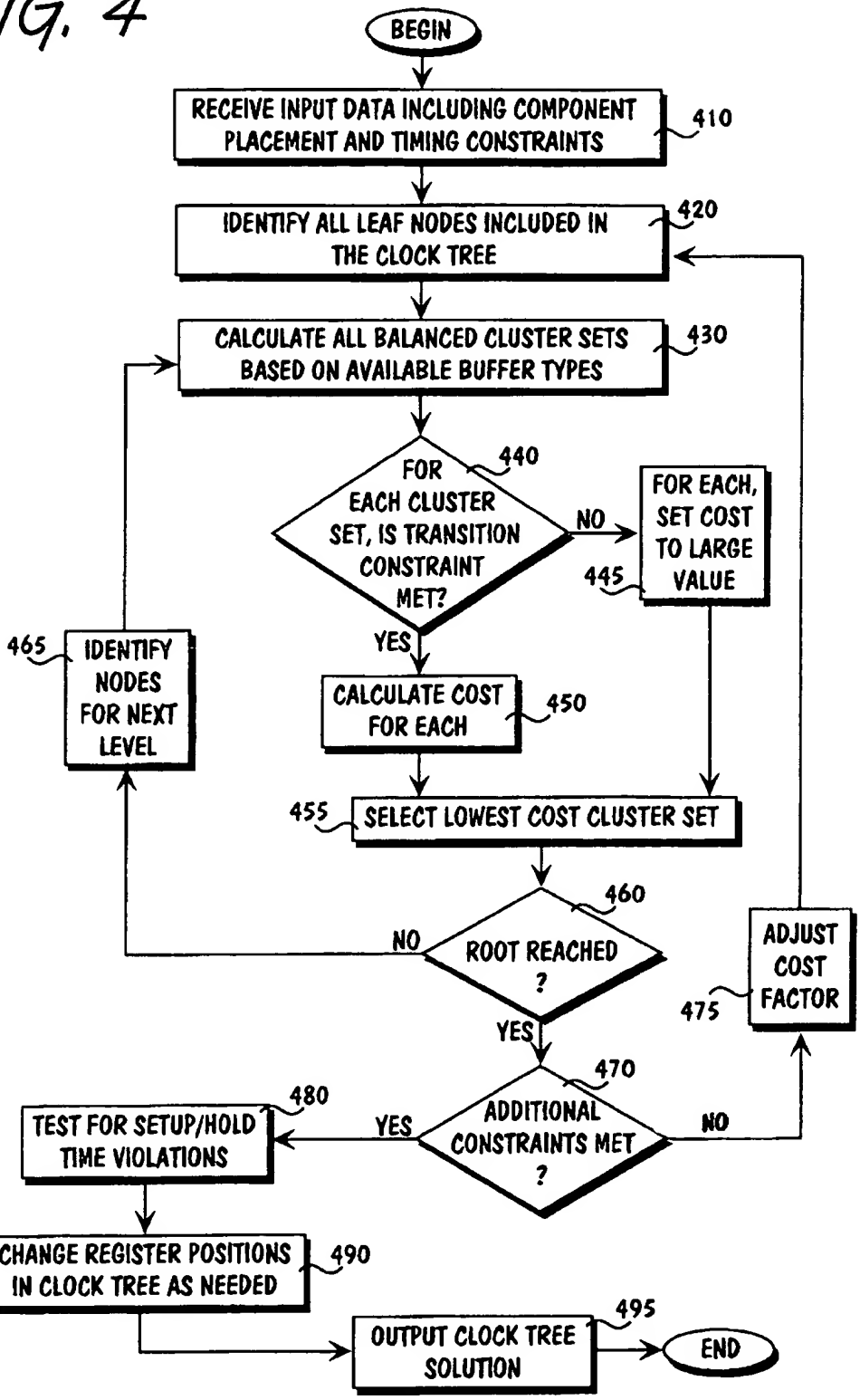

INPUT 330
CLOCK TREE SYNTHESIZER 320
OUTPUT 340
EDA SYSTEM
310

FIG. 3

FIG. 4


BEGIN
RECEIVE INPUT DATA INCLUDING COMPONENT PLACEMENT AND TIMING CONSTRAINTS
410
IDENTIFY ALL LEAF NODES INCLUDED IN THE CLOCK TREE
420
CALCULATE ALL BALANCED CLUSTER SETS BASED ON AVAILABLE BUFFER TYPES
430
FOR EACH CLUSTER SET, IS TRANSITION CONSTRAINT MET?
440
NO
FOR EACH, SET COST TO LARGE VALUE
445
YES
IDENTIFY NODES FOR NEXT LEVEL
465
CALCULATE COST FOR EACH
450
SELECT LOWEST COST CLUSTER SET
455
ROOT REACHED ?
460
NO
YES
ADJUST COST FACTOR
475
ADDITIONAL CONSTRAINTS MET ?
470
YES
NO
TEST FOR SETUP/HOLD TIME VIOLATIONS
480
CHANGE REGISTER POSITIONS IN CLOCK TREE AS NEEDED
490
OUTPUT CLOCK TREE SOLUTION
495
END

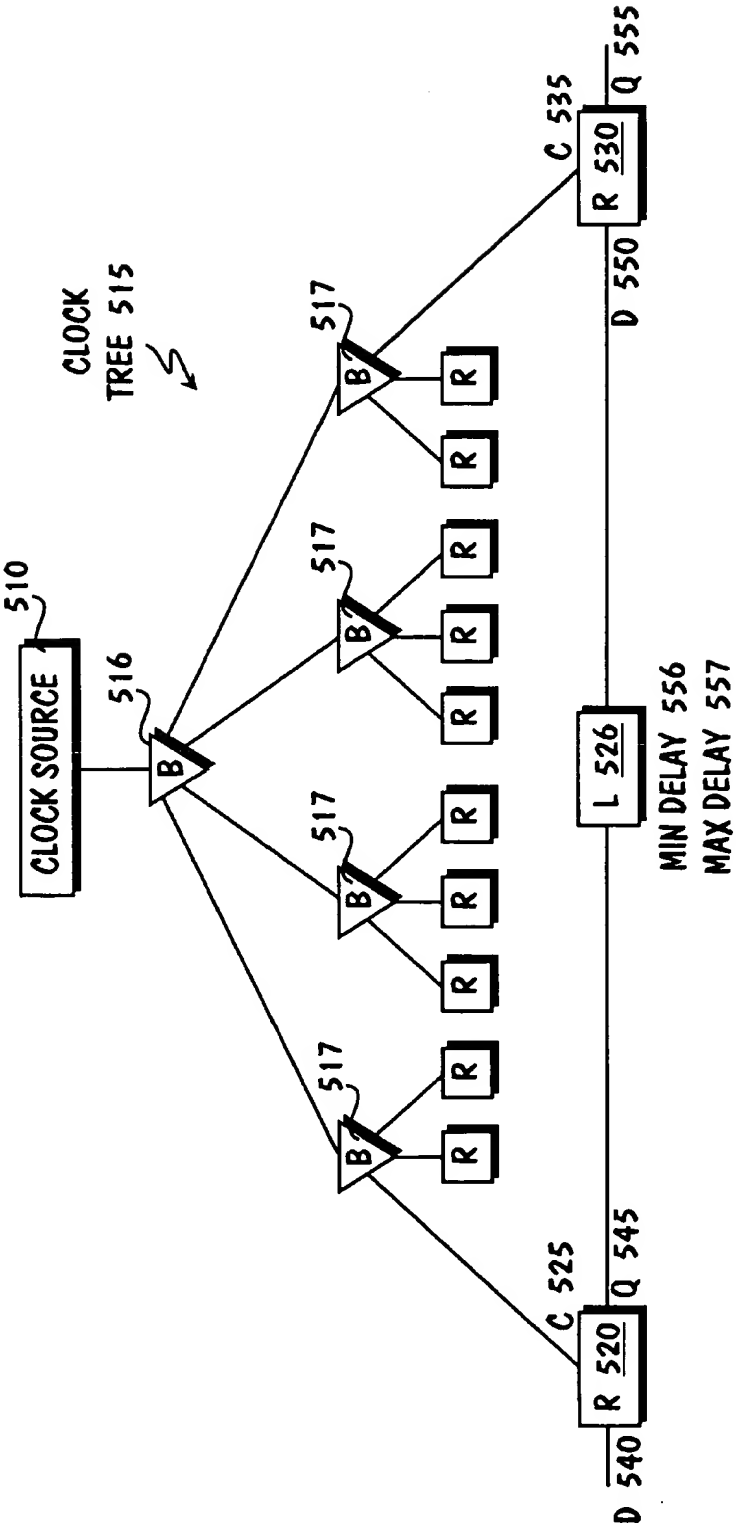

CLOCK TREE 515
510
CLOCK SOURCE
516
B
517
B'
R
C 535
R 530
Q 555
D 550
L 526
MIN DELAY 556
MAX DELAY 557
C 525
R 520
Q 545
D 540

FIG. 5

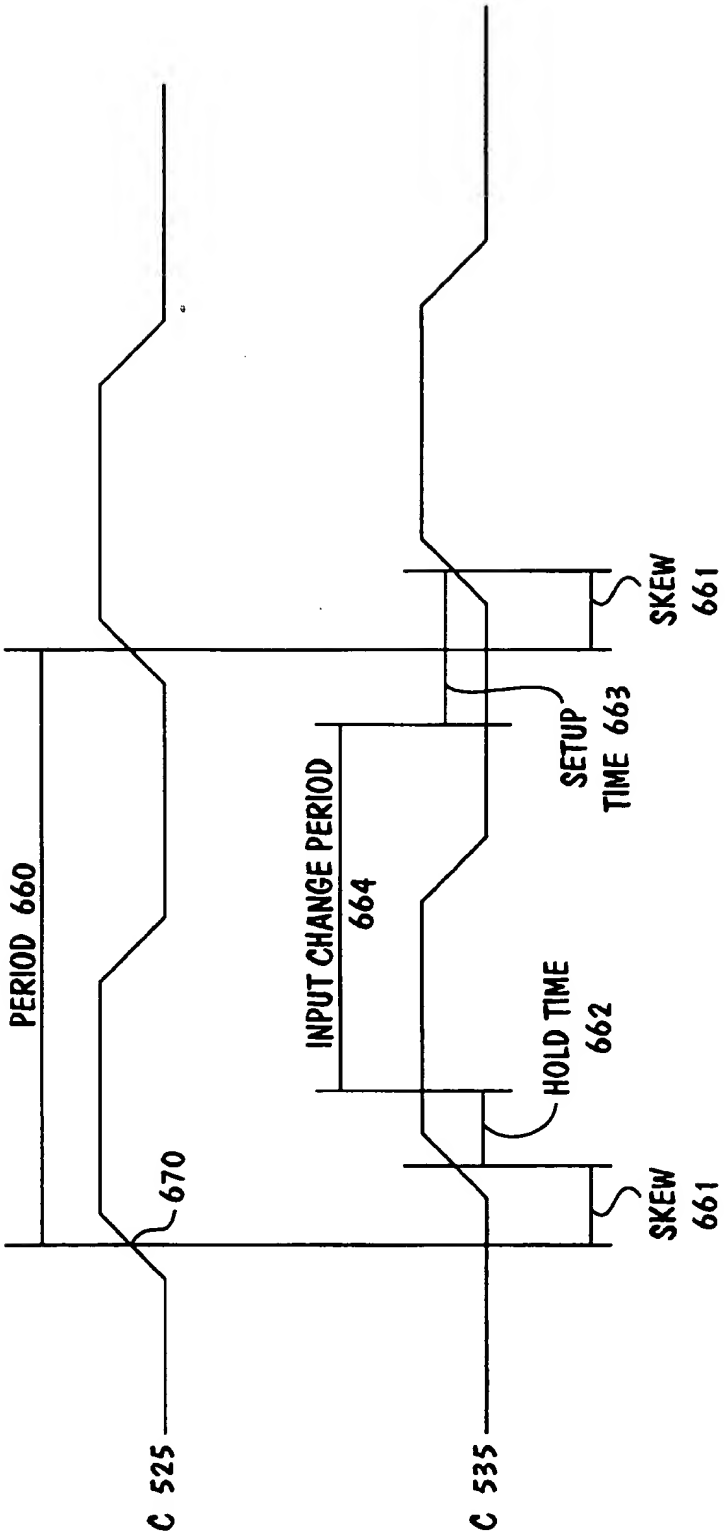

PERIOD 660
670
INPUT CHANGE PERIOD 664
SETUP TIME 663
HOLD TIME 662
SKEW 661
SKEW 661
C 525
C 535

FIG. 6

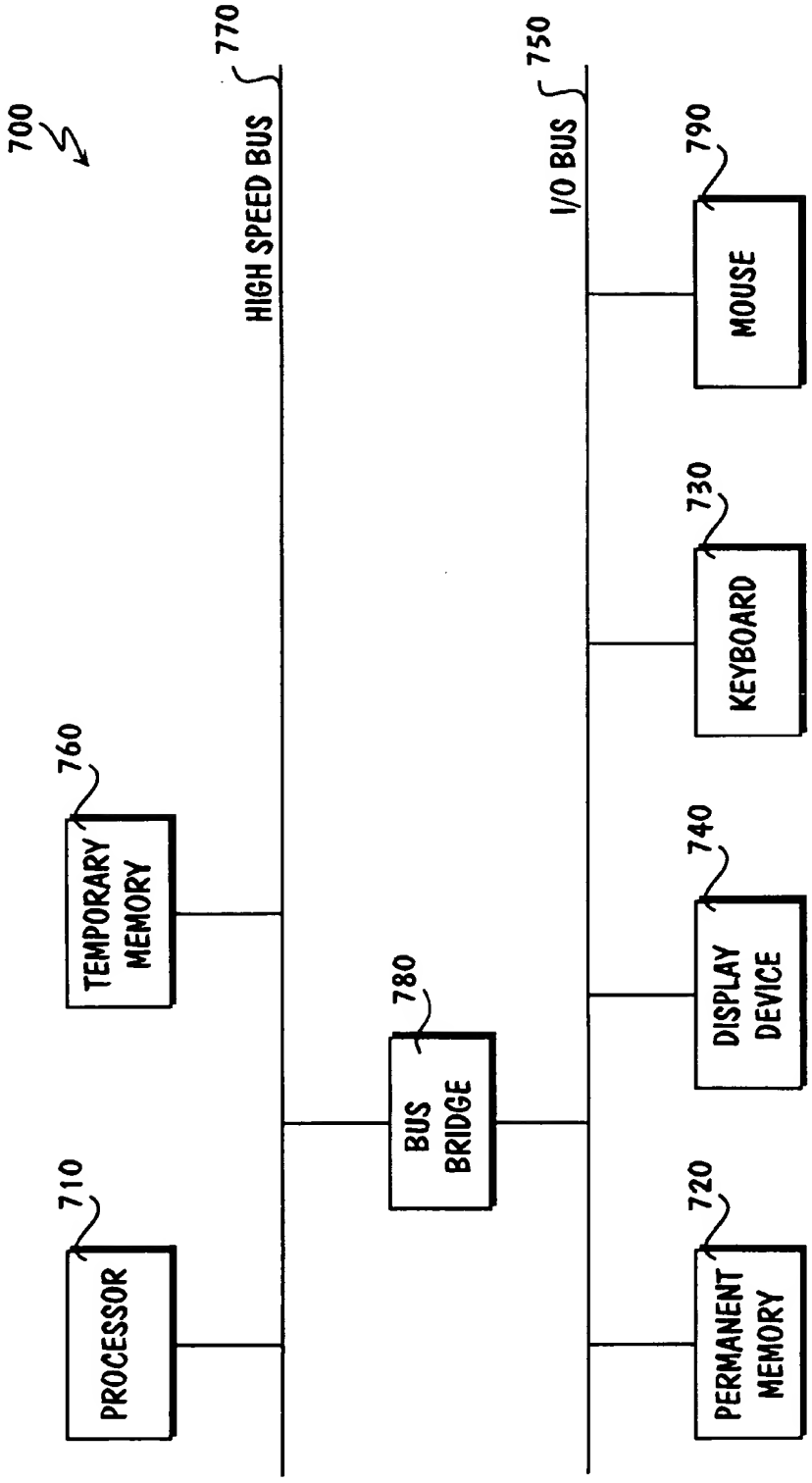

700
770
HIGH SPEED BUS
750
I/O BUS
790
MOUSE
730
KEYBOARD
760
TEMPORARY MEMORY
740
DISPLAY DEVICE
780
BUS BRIDGE
710
PROCESSOR
720
PERMANENT MEMORY

FIG. 7

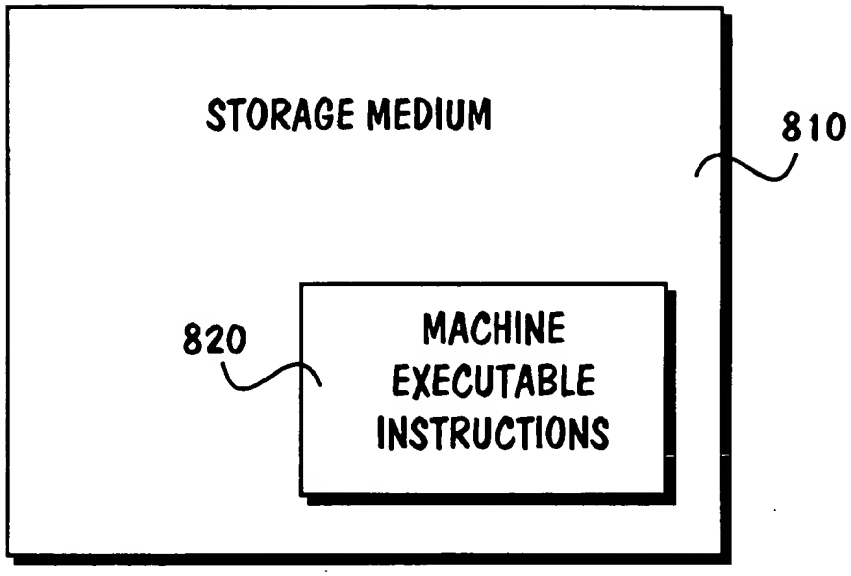


FIG. 8

METHOD AND APPARATUS FOR CLOCK TREE SOLUTION SYNTHESIS BASED ON DESIGN CONSTRAINTS

FIELD OF THE INVENTION

The present invention pertains to the field of integrated circuit (IC) design. More particularly, this invention relates to the art of synthesizing clock tree solutions.

BACKGROUND OF THE INVENTION

Since the advent of the integrated circuit (IC), circuit components have become smaller and smaller. An IC may include millions of components packed into an incredibly small package. With each new generation of smaller integration, more functionality, and therefore more value, can be derived from ICs. Reliably manufacturing these highly integrated ICs, however, presents significant design challenges.

In particular, designing ICs that meet timing constraints can be particularly difficult. An IC may include tens of thousands of registers that need to be connected to one or more clock sources. For each clock "tick", or clock transition, thousands of registers have to operate in concert. A complex network is needed to propagate the clock signal to each of the registers. If the difference in propagation delay through two different paths in the network is too large or too small, errors may occur that can cause the entire IC to fail.

Those skilled in the art will be familiar with numerous processes for synthesizing clock networks, or clock tree solutions. One of the most common approaches is a binary clock tree. A binary clock tree often begins by coupling registers into pairs. Then, pairs of register pairs are coupled together, pairs of pairs of register pairs are coupled together, and so on until the clock source, commonly referred to as the "root" or root node, is reached.

The result is a clock tree having a root and a series of branches reaching out to the registers. The registers are commonly referred to as "leaf nodes" on the tree. Between the root and the leaf nodes there may be several levels of intermediate nodes where paths branch.

Each register and each path adds a certain amount of load to the tree. The root usually cannot drive enough current into the tree to operate the cumulative load. In order to handle large trees, buffers are inserted into the tree at various intermediate nodes. Buffers receive a signal from an upstream driver, such as another buffer or the root node, and drive the signal to a number of down stream nodes.

A wide variety of approaches have been used to insert buffers in clock trees. For instance, the number of nodes coupled to a root may be counted, and one or more buffers inserted as needed. Then, each buffer can be treated like a root in a "sub-tree," and nodes can be counted and buffers inserted to create further sub-trees in a hierarchy that reaches out to the leaf nodes. Various design constraints can be tested, and the process repeated with different types of buffers and tree structures until a suitable solution is found.

As ICs continue to become more complex, having tens of thousand of registers which may be clocked by several different source clocks, at several different clock frequencies, through gated clocks, inverted clocks, etc., the processing time and expense required to meet continually more stringent design constraints using known approaches is becoming increasingly prohibitive.

Therefore, an improved method and apparatus for synthesizing clock tree solutions is needed.

SUMMARY OF THE INVENTION

The present invention beneficially provides an improved method and apparatus for synthesizing clock tree solutions. At a particular level of a clock tree in a circuit description, balanced cluster sets of nodes are calculated based on a set of available buffer types. Each balanced cluster set is tested to see if it meets a design constraint. If the design constraint is not met for a particular balanced cluster set, the particular cluster set is removed from consideration in the clock tree solution. For the cluster sets that do meet the design constraint, a cost associated with each cluster set is calculated. A balanced cluster set that has the lowest cost is selected for the clock tree solution.

In one embodiment, the lowest cost balanced cluster set for one level in the clock tree forms the nodes for the next higher level in the clock tree, and the process is repeated at each level of the clock tree up to a root node. In another embodiment, the entire clock tree is tested to see if it meets a second design constraint. In another embodiment, the clock tree is tested for setup time and/or hold time violations, and register positions within the clock tree are changed to eliminate any violations. In another embodiment, the clock tree in the circuit description is modified with the lowest cost balanced cluster set for each level of the clock tree solution, wherein each cluster includes the buffer on which the cluster calculation was based.

BRIEF DESCRIPTION OF THE DRAWINGS

Examples of the present invention are illustrated in the accompanying drawings. The accompanying drawings, however, do not limit the scope of the present invention. Similar references in the drawings indicate similar elements.

FIG. **1** illustrates one embodiment of an IC design.

FIG. **2** illustrates one embodiment of the IC design with a clock tree solution.

FIG. **3** illustrates one embodiment of the present invention.

FIG. **4** illustrates a process of one embodiment of the present invention.

FIG. **5** illustrates a clock tree for which hold time and setup time violations need to be tested.

FIG. **6** illustrates a timing diagram with a clock skew between clock signals at two registers.

FIG. **7** illustrates one embodiment of a machine used to implement the present invention.

FIG. **8** illustrates one embodiment of a machine readable storage medium to store instructions embodying the present invention.

DETAILED DESCRIPTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, those skilled in the art will understand that the present invention may be practiced without these specific details, that the present invention is not limited to the depicted embodiments, and that the present invention may be practiced in a variety of alternate embodiments. In other instances, well known methods, procedures, components, and circuits have not been described in detail.

Parts of the description will be presented using terminology commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. Also, parts of the description will be presented in terms

of operations performed through the execution of programming instructions. As well understood by those skilled in the art, these operations often take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through, for instance, electrical components.

Various operations will be described as multiple discrete steps performed in turn in a manner that is helpful in understanding the present invention. However, the order of description should not be construed as to imply that these operations are necessarily performed in the order they are presented, or even order dependent. Lastly, repeated usage of the phrase "in one embodiment" does not necessarily refer to the same embodiment, although it may.

The present invention provides an improved method and apparatus for synthesizing clock tree solutions in integrated circuit designs. FIG. **1** illustrates a very simple example of an integrated circuit (IC) design **110** as it may be defined, for instance, by a netlist prior to synthesizing a clock tree solution. Eight registers (R) **120** and four blocks of combinational logic (L) **130** are placed in the circuit design and coupled to each other and to input pins (D1, D2, D3, and D4) and output pins (Q1, Q2, Q3, and Q4) as shown. The netlist also defines a clock tree. Input clock pin (CLK) **140** is coupled to the clock pin of each of the registers **120**. The netlist defines all of the connections but does not define how the connections are made.

CLK **140** is a root node in the clock tree and each register is a leaf node in the clock tree. If CLK **140** cannot drive enough current to operate all eight registers **120**, one or more buffers need to be inserted in the clock tree. In a simple integrated circuit design like the one illustrated in FIG. **1**, buffers could probably be inserted manually, for instance, by modifying the netlist using any of a number of user interfaces. FIG. **2** illustrates IC **110** with a modified clock tree including buffer **210** to drive the clock signal to each of the registers **120**.

Manually modifying clock trees becomes much more difficult when circuits become more complex and design constraints become more stringent. For instance, timing constraints for IC **110** may include minimum and maximum propagation delay from CLK **140** to the registers **120**, minimum and maximum clock transition time at each register **120**, minimum and maximum delay through each logic block **130**, and required setup and hold times for each register **120**. The timing constraints may be very stringent, requiring a balanced solution with very little deviation in delay from one clock path to the next. Additional design constraints may state that CLK **140** can only drive one buffer, buffers that can be used in IC **110** can only drive up to three registers, each buffer introduces a certain amount of propagation delay and increases transition time by a certain amount, and the area available on IC **110** to add buffers is extremely limited, leaving room for no more than four buffers. With these design constraints, manually synthesizing a clock tree solution for even the simple IC design show in FIG. **1** is no trivial matter. For today's highly integrated circuits, often including tens of thousands of registers, manually synthesizing a clock tree solution is virtually impossible.

The present invention beneficially synthesizes clock tree solutions using design constraints and cost analysis to insert buffers and select node clusters that provide superior, balanced clock trees. A clock tree solution can then be added to a circuit description that is later used to route the connections in the IC design.

FIG. **3** illustrates one embodiment of clock tree synthesizer **320** coupled to electronic design automation (EDA) system **310**. Except for the teachings of the present invention, EDA system **310** represents any of a broad category of EDA systems. For instance, EDA system **310** may include capabilities for generating a gate-level IC design from hardware description language (HDL) files, including provision of a timing budget, generation of a floor plan, synthesis of gates, placement of gates, and routing of transmission paths.

EDA system **310** provides clock tree synthesizer **320** with input **330**. Input **330** includes a circuit description in any of a number of formats. In one embodiment, input **330** includes component placement, timing constraints, and a set of available buffers. In response to input **330**, clock tree synthesizer **320** provides output **340** which defines a clock tree solution for the circuit description. In one embodiment, EDA system **310** uses the clock tree solution to route the clock tree in the IC design.

FIG. **4** demonstrates one embodiment of clock tree synthesizer **320**. In general terms, clock tree synthesizer **320** groups clock tree nodes on a level-by-level basis starting with the level of nodes furthest from the root node. At each level, clusters of nodes are calculated in various ways depending on available buffer types, and a best cluster set for each level of the clock tree solution is selected based on a cost analysis. The embodiment illustrated in FIG. **4** includes a number of implementation specific details and various alternate embodiments.

In block **410**, input data is received. In the illustrated embodiment, the input data includes component placement and timing constraints for a circuit design. Component placement includes coordinate locations of clock pins in one or more clock trees in the circuit design, such as source clock locations and the locations of clock inputs on each register, and a definition of component connections. Component placement may be in the form of a netlist.

Timing constraints may include data such as minimum and maximum propagation delay from the source clock to a clock pin of any register, minimum and maximum clock transition time at any register, hold time and setup time requirements for each type of register, and propagation delays through components such as registers, combinational logic, buffers, inverters, clock dividers, clock multipliers, etc. Timing constraints may also include propagation constants for calculating propagation delays through lengths of transmission paths.

In various embodiments, input data may also include design constraints defining one or more source clock wave forms, available area for inserting buffers and/or inverters, available types of buffers and/or inverters including maximum load for each buffer/inverter and area required to insert each buffer/inverter, and available layers in the IC design for clock tree routing.

Input data may also define pre-designed partial trees, also called sub-trees or macro-cells. The partial trees can be treated as a single terminal node from the perspective of the clock tree synthesizer. The input data for partial trees may include maximum and minimum propagation delay from the source clock up to the root of the partial tree, as well as the load the partial tree places on the clock tree.

Certain aspects of the input data can be user defined. For instance, in one embodiment, a default set of available buffers will be used unless a user defined set of buffers is included in the input data. The input data may also indicate certain user defined nodes that should be ignored or are to be

treated as leaf nodes or terminal nodes, as in the case of a partial tree discussed above. Similarly, where two clock trees overlap, a user may be required to define certain nodes to be terminal nodes in order to separate the clock trees as viewed from the perspective of the clock tree synthesizer.

In block **420**, leaf nodes for a given clock tree are identified. For instance, a root node can be selected from a netlist and all of the registers coupled to the root node as defined by the netlist can be identified. The set of identified leaf nodes may include tens of thousands of nodes. The set of leaf nodes comprise the outer most level of the clock tree.

Partial trees, as mentioned above, may be treated as terminal nodes in a clock tree. The timing constraints at the root of a partial tree, however, are likely to be different from the timing constraints at registers. For instance, the propagation delay from the clock tree root to each register in the clock tree must fall within a specified range, but the propagation delay from the clock tree root to the root of a partial tree may not fall within the same specified range. In which case, the partial tree needs to be given special consideration during clock tree synthesis. Partial trees, as well as other types of terminal nodes that are not leaf nodes, will be discussed more fully below.

Continuing on with FIG. **4**, in block **430**, balanced cluster sets of the leaf nodes are calculated based on the available types of buffers. For instance, in one embodiment, the set of available buffers includes five types of buffers that can drive loads up to 15, 10, 5, 3, and 2 registers respectively. In which case, if there are 150 thousand registers at the leaf node level in the clock tree, the registers could be clustered into a first balanced set of 10 thousand clusters of nodes driven by 10 thousand 15-output buffers, or a balanced set of 15 thousand clusters driven by 15 thousand 10-output buffers, or a balanced set of 30 thousand clusters driven by 30 thousand 5-output buffers, and so on for each buffer type.

Each buffer type has associated with it a certain amount of propagation delay and each buffer type impacts clock transition times at the registers by a certain amount. Each set of calculated clusters is "balanced" in that the same buffer type is used for the entire level of leaf nodes so that the timing constraints at each register is similarly impacted. The goal of the ideal clock tree, of course, is for each register to receive the clock signal at exactly the same time so that there is no clock skew between registers. For instance, if the 15-output buffer has a 5 nanosecond propagation delay, then the balanced set of registers all clustered using the 15-output buffer will all experience a 5 nanosecond delay.

Realistically, the total number of nodes will not be evenly divided by the number of nodes that can be driven by a buffer type. Buffers, however, can drive fewer than the maximum number of nodes. In which case, one buffer type can be used for more than one set of clusters. For instance, if there are 145 thousand registers, a cluster set could include 9667 clusters driven by 9667 15-output buffers, all driving 15 registers except one buffer which drives 10 registers. Of course, propagation delay through a buffer may depend on the load, i.e. the number of nodes. In which case, the one 15-output buffer driving only 10 registers may have a significantly shorter propagation delay or transition time, potentially creating clock skew between registers. In which case, in one embodiment, a "balanced" set of clusters is also one that attempts to evenly distribute the number of nodes over the number of buffers. For instance, 145 thousand registers could be driven by 9662 buffers driving 15 registers each and 5 buffers driving 14 registers each. The difference in propagation delay between driving 14 and 15 buffers may be negligible.

In any event, a potentially large number of possible balanced cluster sets are calculated. Any number of techniques can be used to derive the factorization of the number of nodes by the set of available buffers.

In block **440**, each calculated cluster set is tested against a design constraint. In the illustrated embodiment, the tested design constraint is the clock transition time at the registers. For instance, the timing constraints may require that the clock signal at each register must transition from high to low, or low to high, in a minimum of 1 nanosecond and a maximum of 2 nanoseconds. Certain buffer types, driving certain numbers of nodes, may not meet the timing constraint.

Timing constraints are often process dependent, meaning that, in order for the type of register being used in a particular IC design to have a known state, the constraints must be met. If the constraints are not met, errors may occur. In which case, in block **445**, any cluster set that does not meet the timing constraint is removed from consideration. In the illustrated embodiment, a cluster set is effectively removed from consideration by setting a cost associated with the cluster set to a large value so that it will not be selected in the costing analysis discussed below for block **455**.

In block **450**, a cost is calculated for cluster sets that met the timing constraint tested in block **440**. In one embodiment, cost is equal to the cumulative area necessary to insert the set of buffers plus a cost factor times the propagation delay for the buffer type. That is:

$$COST=AREA+\alpha(DELAY)$$

Different buffer types require different amounts of area. The area for each buffer type may be defined in the input data or retrieved from a default library. In general, buffers which drive larger maximum loads require more area. Also, buffers tend to have longer propagation delays for larger loads. A 15-output buffer used to drive only 10 nodes may actually have shorter delay, but require more area, than a 10-output buffer used to drive the same 10 nodes. That is, depending on which component of the cost equation is emphasized based on the value of α, certain cluster sets may have lower associated cost. The cost calculation, and the cost factor α are discussed more fully below. In alternate embodiments, any number of cost equations can be used.

In block **455**, a cluster set having the lowest calculated cost is selected for the clock tree solution. That is, for the given cost equation with a given value for α, the best cluster set for the leaf node level is selected. Since the cluster sets that did not meet the timing constraint in block **440** were set to large values, such as orders of magnitude greater than any reasonable cost value anticipated using the given cost equation, those cluster sets are effectively removed from consideration.

In alternate embodiments, cost is calculated for all cluster sets, not just those that meet the design constraint. In which case, the design constraint may be tested, and cluster sets that do not meet the design constraint removed from consideration, at any point prior to selecting the lowest cost cluster set.

In block **460**, the clock tree synthesizer determines whether or not the root node has been reached. The root node has been reached if all of the nodes in the current level (the leaf node level at this point in the process) are coupled to the root node. Only in very small IC designs will the leaf node level couple directly to the root node. In the example from above, including **150** thousand registers, the leaf node level is many levels removed from the root node.

In block **465**, if the root node has not been reached, the nodes for the next level of the clock tree are identified. The buffers forming the clusters for the previous level comprise the set of nodes for the next level of the clock tree. For instance, if the previous level included 150 thousand leaf nodes, and the lowest cost cluster set was 10 thousand 15-output buffers, the nodes for the next level are the 10 thousand buffers. In which case, the process returns to block **430**, balanced cluster sets are calculated for 10 thousand nodes, and the best cluster set for the 10 thousand-node level of the clock tree solution is selected based on the cost calculation. Levels of the clock tree are built one on top of the other until the root node is reached in block **460**.

The result is a multi-level clock tree solution. Each level is "balanced" in that each level is driven by one type of buffer (or inverter as the case may be). Within each level, in certain embodiments, clusters are also "balanced" in that each cluster includes roughly equally numbers of nodes. The clock signal is propagated to each register through the same number of the same types of buffers.

In block **470**, additional design constraints are tested. In one embodiment, only one additional design constraint is tested—the cumulative propagation delay from the root node to the registers. The propagation delay to every leaf node must be between a minimum and maximum value. Alternate embodiments may test the delay and the cumulative area, or any number of other design constraints or combinations of constraints. If any of the constraints are not met, the clock synthesizer proceeds to block **475**.

In one embodiment the values for the cost equation are cumulated from one level to the next so that, for instance, cumulative delay and area constraints can be tested in block **470** without recalculating the values.

In block **475**, if the additional design constraints are not met, the cost equation is adjusted. In the illustrated embodiment, the cost equation is adjusted by changing the value of the cost factor α. Then the process returns to block **420** and begins to build a new clock tree solution starting at the leaf node level. Changing the cost factor changes the emphasis of the cost equation so that different buffer types and cluster structures are likely to be selected.

In one embodiment, the cost factor is adjusted from one iteration of building a clock tree solution to the next using a binary search until the design constraints are met or no acceptable solution can be found. For instance, a range of acceptable cost factor values may go from zero up to some maximum value. For a first iteration through the process, the cost factor can be set to zero. In which case, recalling the cost equation,

$$COST=AREA+\alpha(DELAY),$$

if the cost factor α is zero, cost will be equal to area. In other words, the first iteration will select the clock tree solution that has the lowest required area without any consideration whatsoever for delay. In one embodiment, at the end of the first iteration, the area and delay constraints can be tested. If the area constraint is not met, then the clock tree may not be physically possible, and the process may end.

For the delay constraint, the propagation delay must be between the minimum and maximum propagation delay. In general, larger cumulative area often translates into shorter overall propagation delay, and longer propagation delay often translates into smaller area. If the propagation delay is longer than the maximum allowable delay, the cost factor should be increased so delay is more emphasized in the cost equation, likely resulting in a larger cumulative area. In the binary search, starting with a zero cost factor in the first iteration, the cost factor can be set to a value half way between zero and the maximum value for the next iteration.

At the end of the next iteration, if the propagation delay is less than the minimum allowable delay, the cost factor should be decreased from the midpoint in the binary search range in order to place more emphasis on the area element of the cost equation, likely resulting in a longer propagation delay. Conversely, if the propagation delay is still more than the maximum allowable delay, the cost factor should be increased again. In which case, the cost factor should be either be decreased to halfway between the current value and zero, or increased to halfway between the current value and the maximum value. With each subsequent iteration, the binary search range gets smaller and smaller because the cost value is either increased by half from the previous value to reduce delay or decreased by half from the previous value to increase delay until an acceptable clock tree solution is found.

Any number of alternate search techniques can be used in alternate embodiments to adjust the cost equation from one iteration to the next. In one alternate embodiment, the design constraints are tested after each level is added to the clock tree solution rather than waiting until the end of a complete iteration.

In block **470**, if the tested design constraints are met, the process proceeds to block **480**. In block **480**, the clock tree is tested for setup time and/or hold time violations. In block **490**, register positions are changed in the clock tree as needed in order to correct any setup time and/or hold time violations. The functions of blocks **480** and **490** are discussed below in more detail with respect FIGS. **5** and **6**.

In block **495**, the clock tree synthesizer outputs the acceptable clock tree solution.

As discussed above, terminal nodes which are not leaf nodes (nodes at registers) require special consideration. The design constraints associated with terminal nodes are often not the same as design constraints associated with leaf nodes. For instance, a terminal node that is a root node for a partial tree, a gated clock, a divided clock, etc., is likely to have different propagation delay constraints. If the acceptable range of propagation delay for a terminal register is longer the acceptable range for registers, any number of techniques can be used to add delay to the terminal node so that the terminal node can fit into the clock tree at the leaf node level and be included in block **420** for the first iteration of the process illustrated in FIG. **4**.

Fitting terminal nodes into the clock tree is more difficult if the range of acceptable delay values is narrower from minimum value to maximum value than for registers, or the maximum allowable delay for a terminal node is shorter than the minimum allowable delay for the register. In one embodiment, both of these cases are addressed by cumulating delay for each level of buffers as they are added to the clock tree. When the cumulative delay of the levels of buffers are approximately equal to the difference in the terminal node delay and the register delay, the terminal node is included in the set of nodes used for the next level of the clock tree solution. That is, in block **465** of FIG. **4**, identifying nodes for the next level of the clock tree solution includes comparing cumulative delay of the current level of the clock tree with the difference between terminal node delay constraints and register delay constraints, and adding terminal nodes to the next level if the constraints match, or match to within a particular deviation range.

FIG. **5** illustrates a simple example of a register for which setup time and hold time violations need to be tested as mentioned above in FIG. **4**, block **480**. Clock source **510** is

coupled to clock tree **515**. Clock tree **515** is "balanced". Clusters sizes based on buffers **517** are equal. Each cluster within a level is driven by the same kind of buffer, buffers **517**. Each register is separated from clock source **510** by the same number and the same kind of buffers, one buffer **516** and one buffer **517**. Ideally, each register receives the clock signal at exactly the same time. Realistically, however, slight process variations from buffer to buffer and slightly different path lengths can result in slight variations in cumulative propagation delays experience by two different registers. The difference in delays is called "skew".

When register **520** receives a rising clock edge at clock input **525**, a value at data input **540** is latched in and passed to output **545**. The value passes through combinational logic **526** and a modified value arrives at the data input **550** of register **530** after a certain amount of delay. The delay from the rising edge at clock input **525** to a value arriving at data input **550** is somewhere between the minimum delay **556** and the maximum delay **557**. When the circuit design is operating properly, the value at data input **550** will be clocked into register **530** at the next rising clock edge. The registers are said to be a "dependent" pair of registers, in which register **520** is independent and register **530** is dependent.

FIG. **6** illustrates one embodiment of a timing diagram for the clock signal received at clock inputs **525** and **535** from FIG. **5**. The difference in the propagation delays from the clock source to the respective registers results in skew **661**. That is, the clock signal at register **530** is slightly behind the clock signal at register **520**. Design constraints for the registers include hold time **662** and setup time **663**. If a value at a data input for a register changes during a setup time before a clock edge or the hold time after a clock edge, the value that appears at the output will be unknown. That is, if the hold time or setup time design constraints are violated, the state of the IC will be unknown. Therefore, the value at data input **550** can only safely change during the period of time labeled **664** in FIG. **6** for a given clock period **660**.

In order to prevent hold time or setup time violations, the minimum delay time **556** in FIG. **5**, which is measured from the rising clock edge **670** in FIG. **6**, must be more than skew **661** plus hold time **662** so that any data change at register **530** happens after hold time **662** and during period **664**. Similarly, the maximum delay **557**, measured from clock edge **670**, must be less than the clock period **660** plus skew **661** minus setup time **663** so that any data change happens before setup time **663** and during period **664**. Solving the equations for skew **661**:

Max delay **557**−Period **660**+setup **663**<skew **661**<min delay **556**−hold **662**.

Hold time or setup time violations can be detected using this condition. Maximum delay **557** minus period **660** plus setup time **663** is usually negative, and minimum delay **556** minus hold time **662** is usually positive. In which case, setup time and hold time violations are usually eliminated by making the magnitude of the skew as small as possible, zero or nearly zero. Any number of additional methods could also be used to test for violations.

If violations are detected, one embodiment of the present invention attempts to correct the violations by changing positions of registers in the clock tree. The placement of the registers in the IC design is not altered. Instead, the points at which the registers are coupled to the clock tree are changed. For instance, skew is partially a result of process dependent variations between buffers. That is, in the example illustrated in FIG. **5**, propagation delay through buffer **517** coupled to register **520** is slightly different from the propagation delay through buffer **517** coupled to register **530**. So, to reduce the skew, both registers should be coupled to the same buffer **517** so that they both experience the same delay through the same buffer.

In general terms, in order to reduce skew, registers can be swapped from cluster to cluster in the clock tree so that a dependent pair of registers share as many common buffers as possible. Basically, this means that a dependent pair of registers should be clustered as low as possible in the clock tree, as near to the leaf node level as possible.

Another approach to reduce hold time and setup time violations within a cluster is to change the order in which a pair of dependent registers are clustered. That is, the independent register should be coupled to the buffer immediately followed by the dependent register. Generally, this will result in a shortest possible variation between transmission paths. FIG. **7** is intended to represent a broad category of computer systems. In FIG. **7**, processor **710** includes one or more microprocessors. Processor **710** is coupled to temporary memory **760** by high speed bus **770**. High speed bus **770** is coupled to Input/Output bus **750** by bus bridge **780**. Permanent memory **720** and Input/Output devices, including display device **740**, keyboard **730**, and mouse **790**, are also coupled to Input/Output bus **750**. In certain embodiments, one or more components may be eliminated, combined, and/or rearranged. A number of additional components may also be coupled to either bus **750** and/or **770** including, but not limited to, another bus bridge to another bus, one or more disk drives, a network interface, additional audio/video interfaces, additional memory units, additional processor units, etc.

Clock tree synthesizer **320**, as shown in FIG. **3**, can be executed by processor **710** as a series or sequence of machine readable instructions or function calls stored, for instance, in permanent memory **720** or temporary memory **760**. Alternately, as shown in FIG. **8**, machine executable instructions **820**, representing the function of clock tree synthesizer **320**, could be stored on distribution storage medium **810**, such as a CD ROM, a digital video or versatile disk (DVD), or a magnetic storage medium like a floppy disk or tape. The instructions could also be downloaded from a local or remote server.

Alternately, the present invention could be implemented in any number of additional hardware machines. For instance, one or more ASICs (application specific integrated circuits) could be endowed with some or all of the functionality of clock tree synthesizer **320**, and inserted into system **700** of FIG. **7** as separate components, or combined with one or more other components.

Thus, an improved method and apparatus for synthesizing clock tree solutions has been described. Whereas many alterations and modifications of the present invention will be comprehended by a person skilled in the art after having read the foregoing description, it is to be understood that the particular embodiments shown and described by way of illustration are in no way intended to be considered limiting. Therefore, references to details of particular embodiments are not intended to limit the scope of the claims.

What is claimed is:

1. A method comprising:

calculating a plurality of balanced cluster sets for a plurality of nodes comprising a first level of a clock tree in a circuit description for consideration as part of a clock tree solution, each balanced cluster set based on one of a set of available buffer types;

testing each of the balanced cluster sets to determine if a first design constraint is met;

removing each of the balanced cluster sets that do not meet the first design constraint from consideration in the clock tree solution;

calculating a cost associated with each of the balanced cluster sets that do meet the first design constraint using a cost formula; and

selecting a lowest cost balanced cluster set for the clock tree solution.

2. The method of claim **1** wherein the lowest cost balanced cluster set comprises a plurality of nodes comprising a next level of the clock tree, the method further comprising:

iteratively repeating the calculating the plurality of balanced cluster sets, testing, removing, calculating the cost, and selecting for each next level of the clock tree up to a root level of the clock tree.

3. The method of claim **1** further comprising:

testing the clock tree to determine if at least one additional design constraint is met;

adjusting a cost factor of the cost formula if the at least one additional design constraint is not met; and

repeating the method if the at least one additional design constraint is not met beginning with a leaf node level for the first level.

4. The method of claim **3** wherein the cost formula comprises an area component and a delay component.

5. The method of claim **4** wherein adjusting the cost factor comprises:

selecting a next value in a binary search of a range of cost factor values, the range of cost factor values to define a relative importance of the delay component in the cost formula.

6. The method of claim **4** wherein the binary search begins with a cost factor value that minimizes the relative importance of the delay component in the cost formula.

7. The method of claim **3** wherein the at least one additional design constraint comprises a minimum and maximum clock delay.

8. The method of claim **1** further comprising:

testing a pair of dependent registers at a leaf node level for setup time violations and/or hold time violations, wherein the pair of dependent registers comprises an independent register and a dependent register; and

changing positions of the independent register and the dependent register in the clock tree until setup time violations and/or hold time violations are eliminated.

9. The method of claim **8** wherein changing positions of the independent register and the dependent register comprises at least one of:

positioning the independent register and the dependent register at a same low level in the clock tree; and

coupling the independent register and the dependent register in a cluster in an order of the independent register followed by the dependent register.

10. The method of claim **1** wherein the first design constraint comprises a minimum and maximum clock transition time.

11. The method of claim **1** wherein removing balanced cluster sets from consideration comprises:

setting a cost associated with each of the balanced cluster sets that do not meet the first design constraint to a large value.

12. The method of claim **1** further comprising:

modifying the clock tree in the circuit description with the lowest cost balanced cluster set selected for the clock tree solution, each cluster including one buffer of the type of buffer on which the calculating was based.

13. The method of claim **1** wherein the set of available buffer types is user defined.

14. The method of claim **2** wherein a number of levels in the clock tree is user defined, and wherein iteratively repeating occurs based on the number of user defined levels.

15. The method of claim **2** further comprising:

comparing a cumulative delay of levels of the clock tree solution with a difference between a delay constraint for a terminal node and a delay constraint for a leaf node; and

including the terminal node in the plurality of nodes comprising the next level of the clock tree based on the comparing.

16. The method of claim **1** wherein the plurality of nodes comprising the first level of the clock tree include a terminal node.

17. The method of claim **16** wherein the terminal node includes one of a root of a partial tree, a user defined node, an input to a logic block, an input to a multiplier, and an input to a divider.

18. The method of claim **1** wherein the set of available buffer types includes inverters.

19. An article of manufacture comprising:

a machine readable storage medium;

the machine readable storage medium having stored thereon machine executable instructions, the execution of the machine executable instructions to implement a method comprising:

calculating a plurality of balanced cluster sets for a plurality of nodes comprising a first level of a clock tree in a circuit description for consideration as part of a clock tree solution, each balanced cluster set based on one of a set of available buffer types;

testing each of the balanced cluster sets to determine if a first design constraint is met;

removing each of the balanced cluster sets that do not meet the first design constraint from consideration in the clock tree solution;

calculating a cost associated with each of the balanced cluster sets that do meet the first design constraint using a cost formula; and

selecting a lowest cost balanced cluster set for the clock tree solution.

20. An apparatus comprising:

first circuitry to calculate a plurality of balanced cluster sets for a plurality of nodes comprising a first level of a clock tree in a circuit description for consideration as part of a clock tree solution, each balanced cluster set based on one of a set of available buffer types;

second circuitry to test each of the balanced cluster sets to determine if a first design constraint is met;

third circuitry to remove each of the balanced cluster sets that do not meet the first design constraint from consideration in the clock tree solution;

fourth circuitry to calculate a cost associated with each of the balanced cluster sets that do meet the first design constraint using a cost formula; and

fifth circuitry to select a lowest cost balanced cluster set for the clock tree solution.

* * * * *

US005686845A

United States Patent [19]

Erdal et al.

[11] Patent Number: 5,686,845

[45] Date of Patent: *Nov. 11, 1997

[54] **HIERARCHIAL CLOCK DISTRIBUTION SYSTEM AND METHOD**

[75] Inventors: **Apo C. Erdal**, Saratoga; **Trung Nguyen**, San Jose; **Kwok Ming Yue**, Fremont, all of Calif.

[73] Assignee: **LSI Logic Corporation**, Milpitas, Calif.

[*] Notice: The term of this patent shall not extend beyond the expiration date of Pat. No. 5,570,045.

[21] Appl. No.: **704,831**

[22] Filed: **Aug. 28, 1996**

(Under 37 CFR 1.47)

Related U.S. Application Data

[63] Continuation of Ser. No. 482,763, Jun. 7, 1995, Pat. No. 5,570,045.

[51] **Int. Cl.**[6] **H03K 19/00**

[52] **U.S. Cl.** **326/93**; 327/297; 327/293; 326/101

[58] **Field of Search** 326/93, 101; 327/165, 327/166, 293, 295, 297

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,104,330	9/1963	Hamilton	327/297
4,769,558	9/1988	Bach	327/297
4,812,684	3/1989	Yamagiwa et al.	326/101
4,912,340	3/1990	Wilcox et al. .	
4,926,066	5/1990	Maini et al.	326/101
5,109,168	4/1992	Rusu	326/47
5,140,184	8/1992	Hamamoto et al.	326/93
5,172,330	12/1992	Watanabe et al.	327/297
5,204,559	4/1993	Deyhimy et al. .	
5,206,861	4/1993	Hannon et al.	371/22.3
5,239,215	8/1993	Yamaguchi .	
5,254,955	10/1993	Saeki et al. .	
5,258,660	11/1993	Nelson et al. .	
5,278,466	1/1994	Honoa et al. .	
5,307,381	4/1994	Ahuja	375/107
5,309,035	5/1994	Watson, Jr. et al. .	
5,430,397	7/1995	Itoh et al.	327/297
5,570,045	10/1996	Erdal et al.	326/93

Primary Examiner—David R. Hudspeth
Assistant Examiner—Richard Roseen
Attorney, Agent, or Firm—Oppenheimer Poms Smith

[57] **ABSTRACT**

A microelectronic circuit includes a plurality of circuitry blocks and sub-blocks, a clock driver, an electrical interconnect that directly connects the clock driver to the sub-blocks, and balanced clock-tree distribution systems provided between the electrical interconnect and circuitry in the sub-blocks respectively. A method of producing a hierarchial clock distribution system for the circuit includes determining clock skews between the clock driver and the sub-blocks respectively. Delay buffers are selected from a predetermined set of delay buffers having the same physical size and different delays, with the delay buffers being selected to provide equal clock skews between the clock driver and the distribution systems respectively. Each delay buffer includes a delay line, and a number of loading elements that are connected to the delay line, with the number of loading elements being selected to provide the required clock delay for the respective sub-block.

16 Claims, 3 Drawing Sheets

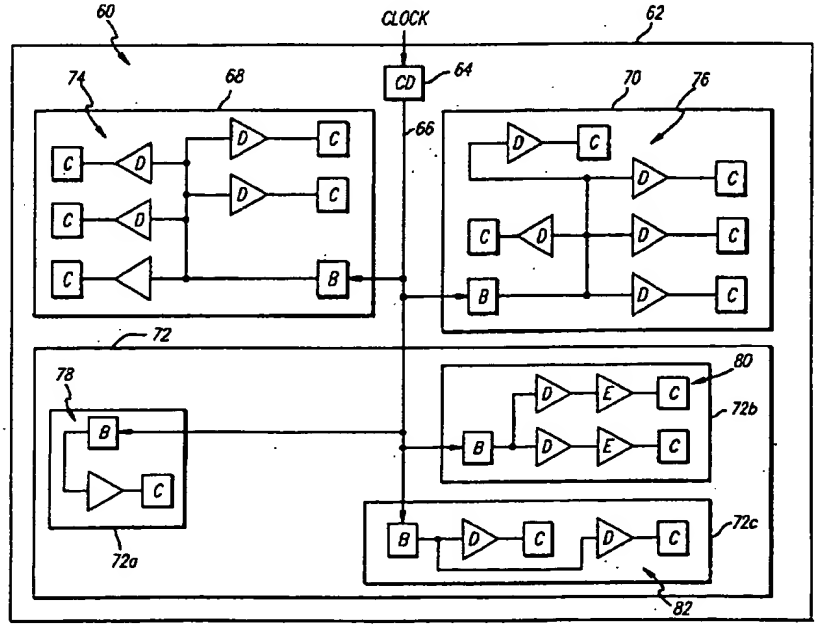


FIG. 1

PRIOR ART

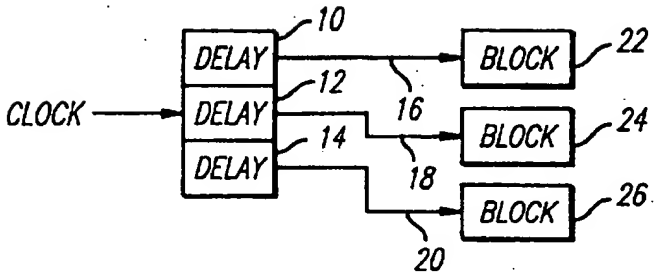


FIG. 2

PRIOR ART

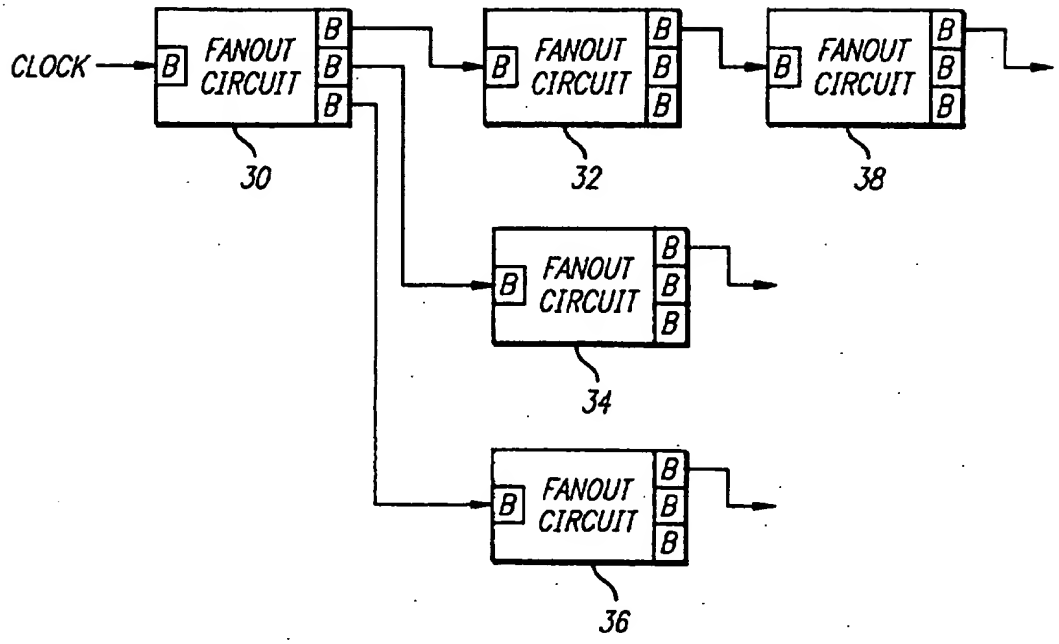


FIG. 3

PRIOR ART

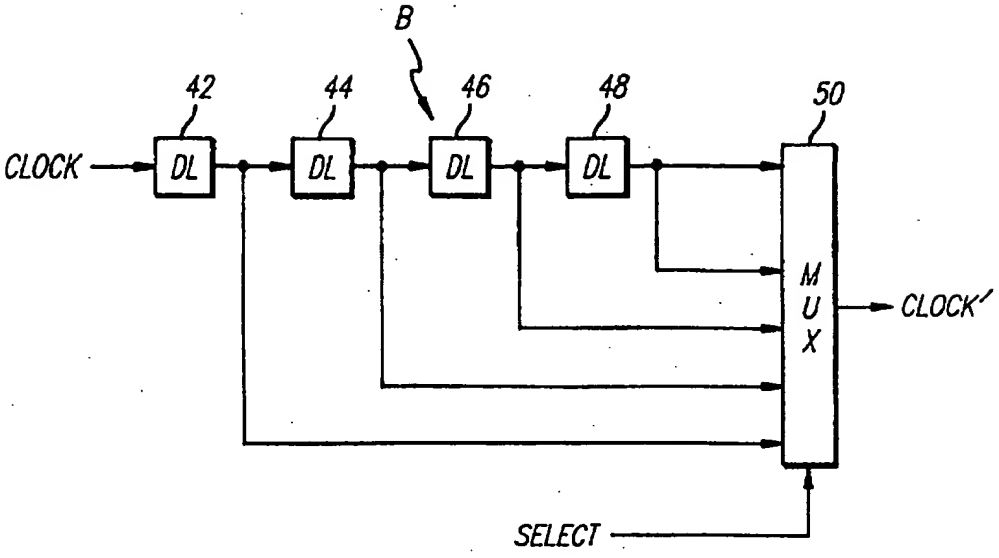
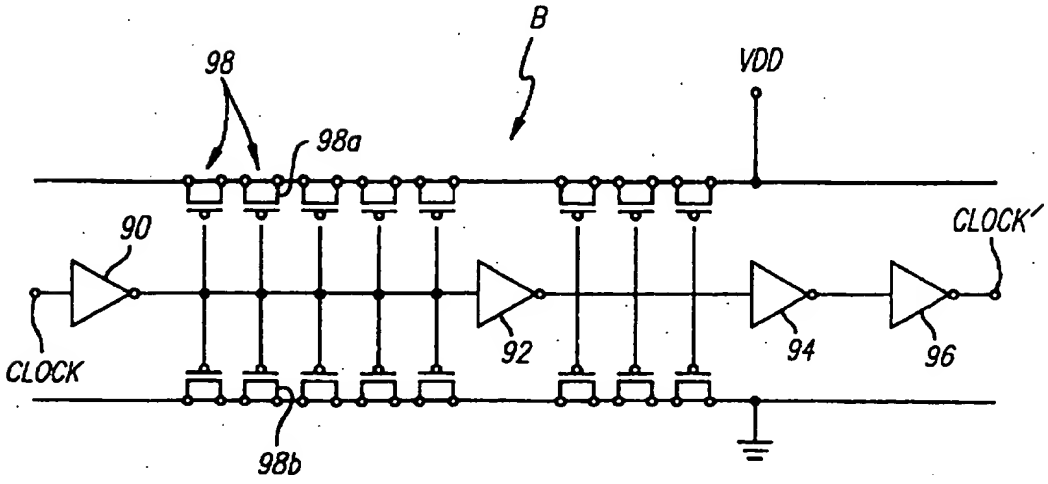


FIG. 5



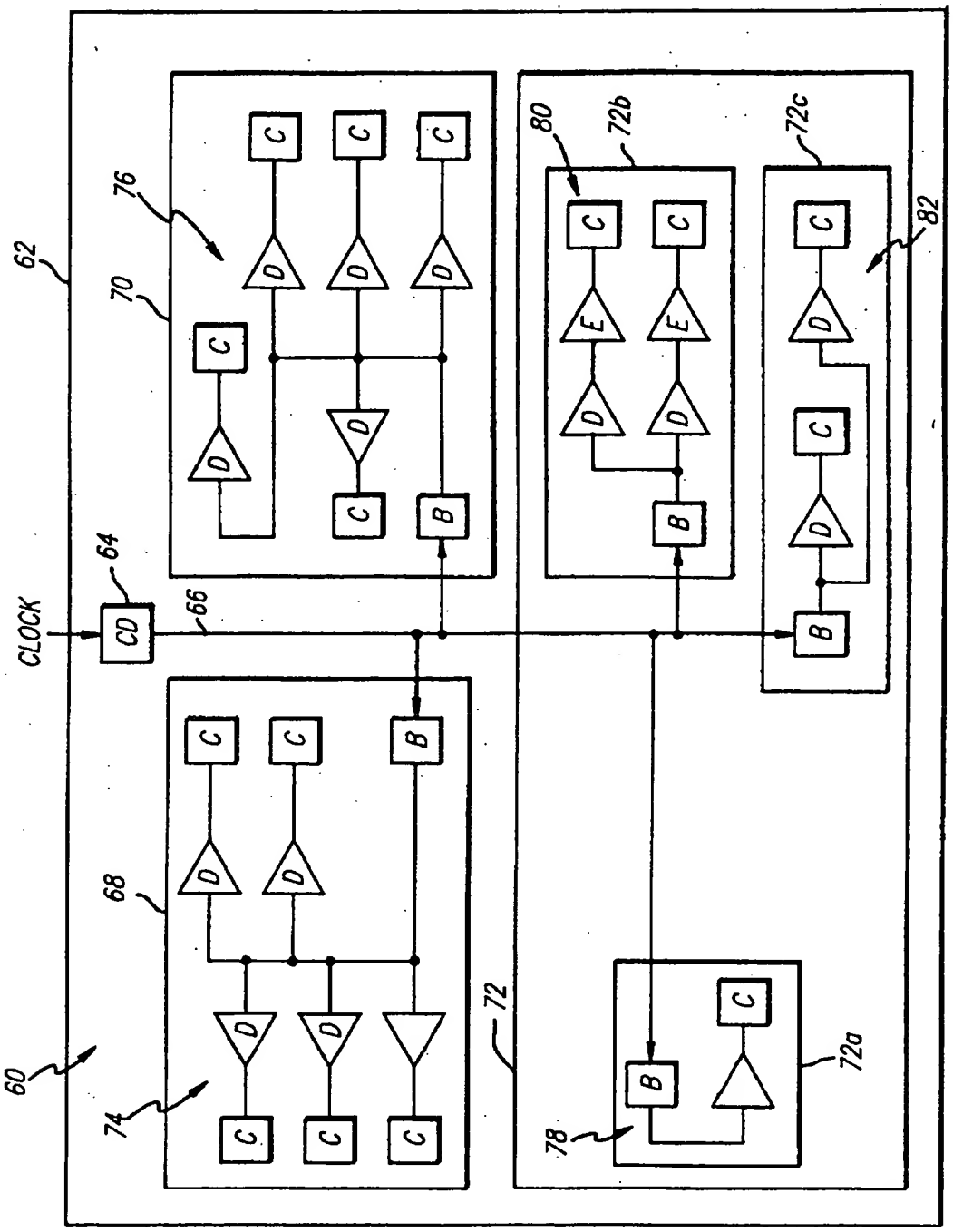


FIG. 4

HIERARCHIAL CLOCK DISTRIBUTION SYSTEM AND METHOD

This application is a continuation of Ser. No. 08/482,763 filed Jun. 7, 1995, now U.S. Pat. No. 5,570,045.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to the art of microelectronic integrated circuits, and more specifically to a hierarchial clock distribution system and method for optimally equalizing clock skew to circuitry in blocks of an integrated circuit.

2. Description of the Related Art

A large microelectronic integrated circuit, such as an Application Specific Integrated Circuit (ASIC), generally includes a number of circuitry blocks or modules that can themselves include sub-blocks in a hierarchial arrangement. The circuitry is driven by clock pulses that are applied through an input clock driver and distributed via interconnect wiring to the various blocks of the circuit and other devices that are not included in the blocks.

In order for the circuit to function properly, the clock pulses must arrive at each clocked circuit element at the same time. However, the lengths of the wiring that conduct the clock pulses to the different blocks will generally be different. Since the length of time required for an electrical signal to propagate through a wire is proportional to the length of the wire, the clock pulses will arrive at the blocks at different times.

In addition, different types of buffers may be used in each block, creating differences between the clock pulse arrival to the clocked circuit elements in the blocks. The phase or timing difference between the clock pulse arrival time to any two clocked circuit element in a microelectronic integrated circuit is called skew.

It is therefore necessary to provide means for minimizing the skew in the circuit and restore synchronism to the operation of the circuit. This function can be provided by inserting delay buffers in the circuit having different delays to compensate for the different values of delay at the individual blocks.

A typical prior art skew compensation system is disclosed in U.S. Pat. No. 5,307,381, entitled "SKEW-FREE CLOCK SIGNAL DISTRIBUTION NETWORK IN A MICROPROCESSOR", issued Apr. 26, 1994 to B. Ahuja. A simplified diagram illustrating this system is presented in FIG. 1.

Input clock pulses CLOCK are applied to a plurality of delay buffers **10**, **12** and **14**, which are connected through lines **16**, **18** and **20** to circuit blocks **22**, **24** and **26** respectively. The buffers **10**, **12** and **14** delay the clock pulses by different lengths of time to compensate for the different lengths of the lines **16**, **18** and **20** such that the clock pulses CLOCK arrive at the blocks **22**, **24** and **26** simultaneously.

An extension of this concept to a hierarchial structure of circuitry blocks is disclosed in U.S. Pat. No. 5,258,660, entitled "SKEW-COMPENSATED CLOCK DISTRIBUTION SYSTEM", issued Nov. 2, 1993 to S. Nelson et al. A simplified diagram of this system is presented in FIG. 2.

The system comprises a plurality of fanout circuits **30**, **32**, **34**, **36** and **38**, each including an input delay buffer and a plurality of output delay buffers which are collectively designated by the reference character B. As illustrated, each fanout circuit has three outputs, although the actual number of outputs is not relevant.

The output delay buffers B of the fanout circuit **30** are connected to the input delay buffers B of the fanout circuits **32**, **34** and **36**, which collectively produce nine outputs. Each output of the fanout circuits **32**, **34** and **36** is connected to an input of another fanout circuit to provide a total of 27 outputs, and the hierarchial chain can continue to as many levels as desired. Only one fanout circuit **38** is illustrated as being connected to an output buffer B of the fanout circuit **32** for simplicity of illustration.

The output buffers of the fanout circuits are connected to respective circuit blocks of a microelectronic integrated circuit, although not explicitly illustrated, in the manner described above with reference to FIG. **1**. Thus, the skew from the clock input to the blocks can be equalized.

However, it is difficult to predetermine and accurately equalize skew using fixed delay buffers in the arrangement of FIG. **2** because any inaccuracy is passed to downstream fanout circuits in the chain. For this reason, the buffers B are implemented as programmable delay elements such as illustrated in FIG. **3**.

Each delay buffer B includes a chain of cascaded fixed delay elements **42**, **44**, **46** and **48** having outputs connected to inputs of a multiplexer **50**. The delay at the output of each delay element is equal to the delay that itself produces plus the accumulated delay of the upstream delay elements. The output of the delay element **42** has a minimum delay value, whereas the output of the delay element **48** has a maximum delay value.

Although not explicitly illustrated, the system comprises a phase locked loop or other type of phase comparator that compares reference clock pulses having the required skew with output pulses CLOCK' from the multiplexer **50** of each buffer B. The comparator then generates and applies a unique SELECT signal to the multiplexer **50** of each buffer B designating which multiplexer input (output of respective delay element **42**, **44**, **46** or **48**) to pass therethrough as output pulses CLOCK'. The value of the SELECT signal corresponds to the delay required to make the phase or skew of the pulses CLOCK' coincide with that of the reference pulses.

Although effective in equalizing the skew in an integrated circuit having a hierarchial block structure, the arrangement of FIGS. **1** to **3** is disadvantageous in that it requires programmable delay buffers and phase comparison circuitry. This undesirably increases the complexity and cost of the integrated circuit.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a hierarchial clock distribution system and method for a microelectronic integrated circuit that enables accurate clock delay compensation using fixed delay buffers to minimize the skew.

A microelectronic circuit includes a plurality of circuitry blocks and sub-blocks, a clock driver, an electrical interconnect that directly connects the clock driver to the sub-blocks, and balanced clock-tree distribution systems provided between the electrical interconnect and circuitry in the sub-blocks respectively.

A method of providing a hierarchial clock distribution system for the circuit includes determining clock delays between the clock driver and the clocked circuit elements within sub-blocks respectively. Delay buffers are selected from a predetermined set of fixed delay buffers having the same physical size and different delays, with the delay buffers being selected to provide equal clock delay between the clock driver and the distribution systems respectively.

Each delay buffer includes a delay line, and a number of loading elements that are connected to the delay line, with the number of loading elements being selected to provide the required clock delay for the respective sub-block.

These and other features and advantages of the present invention will be apparent to those skilled in the art from the following detailed description, taken together with the accompanying drawings, in which like reference numerals refer to like parts.

DESCRIPTION OF THE DRAWINGS

FIG. 1 is a simplified diagram illustrating a prior art system for clock skew equalization;

FIG. 2 is a simplified diagram illustrating how the system of FIG. 1 can be extended to a hierarchial circuitry block arrangement;

FIG. 3 is a diagram illustrating a programmable delay buffer of the system of FIG. 2;

FIG. 4 is a diagram illustrating a microelectronic integrated circuit including a hierarchial clock distribution system embodying the present invention; and

FIG. 5 is an electrical schematic diagram illustrating a fixed clock delay buffer of the system of FIG. 4.

DETAILED DESCRIPTION OF THE INVENTION

A hierarchial clock distribution system embodying the present invention is illustrated in FIG. 4 and generally designated by the reference numeral **60**. The system **60** is implemented as part of a microelectronic integrated circuit **62** which typically receives clock pulses CLOCK from an external source. However, it is within the scope of the invention, although not explicitly illustrated, to provide a clock pulse generator as part of the circuit **62** itself.

The clock pulses CLOCK are applied to a clock driver **64**, which applies the clock pulses through an electrical interconnect wiring **66** to microelectronic circuit modules or blocks **68**, **70** and **72**. The block **72** comprises sub-blocks **72***a*, **72***b* and **72***c*.

The wiring **66** is connected to a clock delay buffer B in each block **68** and **70**, and in each sub-block **72***a*, **72***b* and **72***c*. It will be noted that no buffer B is provided in the block **72** between the wiring **66** and the sub-blocks **72***a*, **72***b* and **72***c*.

Although only two levels of hierarchy are illustrated in FIG. **4**, consisting of one block level and one sub-block level, the invention is not so limited. A hierarchial structure including any number of block/sub-block levels can be provided in accordance with the present invention. However, a delay buffer B will be provided typically at the first and second hierarchial block level.

The individual time delays provided by the delay buffers B are selected to equalize the clock delay between the driver **64** and clocked circuit elements or cells C in each block. Skew compensation within the blocks or sub-blocks **68**, **70**, **72***a*, **72***b* and **72***c* is provided by balanced clock tree distribution systems **74**, **76**, **78**, **80** and **82** respectively.

The specific arrangement of each distribution system will depend on the circuitry in the respective block or sub-block. For simplicity of illustration, each balanced clock tree distribution system **74**, **76**, **78**, **80** and **82** is shown as comprising the clocked circuitry elements or cells C, and local buffers or drivers D and E that are connected between the buffers B and the cells C.

Balanced clock tree distribution is known in the art per se, and the details thereof are not the particular subject matter of the present invention. A basic treatise on this subject is presented in an article entitled "AN IMPLEMENTATION OF A CLOCK-TREE DISTRIBUTION SCHEME FOR HIGH-PERFORMANCE ASICS", by A. Erdal et al, in Proceedings of the Annual IEEE International ASIC Conference and Exhibit, Rochester, N.Y., September 1992, pp. 26 to 29.

In general, balanced clock tree distribution is performed by using a clock partition to split the original clock net into buffered sub-clock nets in a bottom-up fashion after placement of the cells C, without the local buffers D. An initial grouping of clocked cells C (cells having clock pins) is obtained according to the spread of the cells C and the distance among the neighboring clocked cells C. Then, the clocked cells C are exchanged among the groups to get the optimum result.

The objective is to minimize both the maximum absolute loading difference among the groups and the standard deviation of the loading of the groups. After the grouping, an appropriate number of balance cells (not shown) are added to balance the loading for each group.

The location of each balance cell is calculated to balance the area and spread of the group with respect to other groups. The location of the local buffers D and E are calculated as the optimum balance center, based on the estimated routing pattern of the group to minimize the skew among the clocked cells C. Finally, all the balance cells and buffers D and E are inserted into the design and placed automatically in layout, based on the calculated coordinates.

The present invention enables effective clock skew compensation using fixed delay buffers B, which is difficult to obtain using the prior art arrangements illustrated in FIGS. **1**, **2** and **3**. The invention accomplishes this goal by providing the delay buffers B only at one sub-block level, thereby eliminating accumulated hierarchial delay inaccuracies, and performing block level skew compensation using the balanced clock tree distribution systems.

As illustrated in FIG. **5**, each delay buffer B has the same physical size in order to facilitate placement in the circuit **62**. Each buffer B comprises the same number, here illustrated as four, of logic elements which provide known delays. As shown in FIG. **5**, the logic elements are inverters **90**, **92**, **94** and **96**, although the invention is not so limited. The inverters can be replaced by, for example, NOR gates or wire delay lines, although not explicitly illustrated.

The inverters **90**, **92**, **94** and **96** are connected in a cascade or chain, such that the clock pulses CLOCK' are delayed by the sum of the delays provided by the individual inverters **90**, **92**, **94** and **96**. For example, each inverter provides a delay of 0.25 ns, such that the total delay provided by the inverters **90**, **92**, **94** and **96** alone is 1.0 ns.

Each delay buffer B is capable of providing a delay ranging from 1.0 ns to, for example 3.0 ns, by variably loading the outputs of the inverters **90**, **92**, **94** and **96** using loading elements **98**. Each loading element **98** comprises, in the illustrated example, a PMOS field-effect transistor **98***a* and/or an NMOS field-effect transistor **98***b* that have their gates connected to the output of the respective inverter. The source and drain of each transistor **98***a* is connected to a first constant electrical potential source VDD, whereas the source and drain of each transistor **98***b* are connected to a second constant electrical potential source which is illustrated as being ground.

Each loading element **98** causes the delay of the respective inverter **90**, **92**, **94** and **96** to be increased by, for

example, 0.1 ns. A number of delay elements **98** from 0 to 5 can be connected to the output of each inverter **90**, **92**, **94** and **96**, such that the total delay can be increased by 4 inverters×5 loading elements/inverter×0.1 ns delay/loading element=2.0 ns. Thus, the total maximum delay that can be provided by each delay buffer B is 1.0 ns+2.0 ns=3.0 ns, and the delay can be varied from 1.0 ns to 3.0 ns in 20 increments of 0.1 ns/increment.

In configuring a particular buffer B, loading elements **98** are first provided at the output of the first inverter **90**. If more loading elements **98** are required, they are provided at the outputs of the inverters **92**, **94** and **96** in consecutive order.

The design of the integrated circuit **62** is facilitated by providing a circuit library set of 20 delay buffers **98** which differ from each other only in that they have 20 different numbers of loading elements **98** to provide the 20 different values of delay respectively.

A timing analysis is performed on the circuit **62** to determine the value of delay between the clock driver **64** and each clocked cell C. This is accomplished by assigning initially a minimum delay value (1.0 ns) to each delay buffer B, and determining the delay at the input pin of clocked cell C. The timing analysis can be advantageously performed using, for example, the Timing Analyzer Release 2.2 which is commercially available as part of the Concurrent MDE® Design System (C-MDE® Design System) from LSI Logic Corporation of Milpitas, Calif.

After the delay corresponding to each clocked cell C is determined, the value of delay which the buffer B is required to produce in order to equalize the delay at the inputs of all of the clocked cells C is calculated, and one of the 20 possible buffer configurations is selected from the library set which has the corresponding value of delay. The buffers B are then inserted into the design and placed automatically in layout, based on the required delay values.

Various modifications will become possible for those skilled in the art after receiving the teachings of the present disclosure without departing from the scope thereof.

For example, the numbers of delay elements and loading elements in the buffers B as described above, as well as the specific delays that they provide, are exemplary only, and can be varied in any manner to accommodate a particular application.

We claim:

1. A hierarchial clock distribution system for a microelectronic circuit including a plurality of circuitry blocks and sub-blocks, comprising:

a clock driver;

delay buffers provided in the sub-blocks respectively;

an electrical interconnect that directly connects the clock driver to the delay buffers; and

balanced clock-tree distribution systems provided between the delay buffers and circuitry in the sub-blocks respectively;

the delay buffers providing equal clock skews from the clock driver to the distribution systems respectively.

2. A system as in claim **1**, in which the delay buffers have the same physical size.

3. A hierarchial clock distribution system for a microelectronic circuit including a plurality of circuitry blocks and sub-blocks, comprising:

a clock generator;

delay buffers provided in the sub-blocks respectively;

an electrical interconnect that directly connects the clock generator to the delay buffers; and

balanced clock-tree distribution systems provided between the delay buffers and circuitry in the sub-blocks respectively;

the delay buffers providing equal clock skews from the clock generator to the distribution systems respectively,

wherein the delay buffers have the same physical size and comprise identical delay lines that are loaded to equalize said clock skews respectively.

4. A system as in claim **3**, further comprising a clock driver connected between said clock generator and said electrical interconnect.

5. A system as in claim **3**, further comprising a plurality of loading elements, in which:

the delay lines comprise strings of logic elements; and

the loading elements are connected to outputs of the logic elements.

6. A system as in claim **5**, in which each logic element comprises an inverter.

7. A system as in claim **5**, in which each logic element comprises a NOR gate.

8. A system as in claim **5**, in which each logic element has a number n of loading elements connected to the output thereof, where $0 \leqq n \leqq N$, and N is a predetermined maximum value.

9. A system as in claim **8**, in which N=5, each of said loading elements increases the delay by about 0.1 ns, and there are four of said logic elements each having a delay of about 0.25 ns, whereby the delay can be varied from about 1 ns to about 3 ns in 20 increments of about 0.1 ns/increment.

10. A clock delay buffer for a microelectronic circuit, comprising:

a delay line comprising a string of logic elements; and

a number of loading elements that are connected to the delay line, said number being selected to provide a predetermined clock delay, wherein said loading elements comprise MOS-type field-effect transistors having gates connected to outputs of the logic elements, and sources and drains connected to a constant electrical potential.

11. A buffer as in claim **10**, in which each loading element comprises:

a PMOS field-effect transistor having a gate connected to an output of one of the logic elements, and a source and a drain connected to a constant electrical potential.

12. A buffer as in claim **10**, in which each loading element comprises:

a NMOS field-effect transistor having a gate connected to an output of said one of said logic elements, and a source and a drain connected to a constant electrical potential.

13. A buffer as in claim **10**, in which each logic element comprises an inverter.

14. A buffer as in claim **10**, in which each logic element comprises a NOR gate.

15. A buffer as in claim **10**, in which each logic element has a number n of loading elements connected to the logic element, where $0 \leqq n \leqq N$, and N is a predetermined maximum value.

16. A buffer as in claim **15**, in which N=5, each of said loading elements increases the delay by about 0.1 ns, and there are four of said logic elements each having a delay of about 0.25 ns, whereby the delay can be varied from about 1 ns to about 3 ns in 20 increments of about 0.1 ns/increment.

* * * * *

US005564022A

United States Patent [19]

Debnath et al.

[11] Patent Number: 5,564,022

[45] Date of Patent: Oct. 8, 1996

[54] METHOD AND APPARATUS FOR AUTOMATICALLY INSERTING CLOCK BUFFERS INTO A LOGIC BLOCK TO REDUCE CLOCK SKEW

[75] Inventors: Goutam Debnath; Kelly J. Fitzpatrick, both of Beaverton, Oreg.

[73] Assignee: Intel Corporation, Santa Clara, Calif.

[21] Appl. No.: 193,789

[22] Filed: Feb. 9, 1994

[51] Int. Cl.[6] H01L 27/04

[52] U.S. Cl. 395/250; 395/550; 364/239; 364/239.7; 364/284.1; 307/465

[58] Field of Search 395/250, 550; 326/38, 85; 327/158, 141; 364/491

[56] References Cited

U.S. PATENT DOCUMENTS

5,045,725	9/1991	Sasaki et al.	326/37
5,087,829	2/1992	Ishibashi et al.	327/152
5,118,975	6/1992	Hillis et al.	327/158
5,172,330	12/1992	Watanabe et al.	364/491
5,204,559	4/1993	Deyhimy et al.	327/232
5,258,660	11/1993	Nelson et al.	327/141
5,272,390	12/1993	Watson, Jr. et al.	307/269
5,272,729	12/1993	Bechade et al.	375/118
5,298,866	3/1994	Kaplinsky	328/155
5,307,381	4/1994	Ahuja	375/107
5,369,640	11/1994	Watson et al.	371/1
5,397,943	3/1995	West et al.	326/39

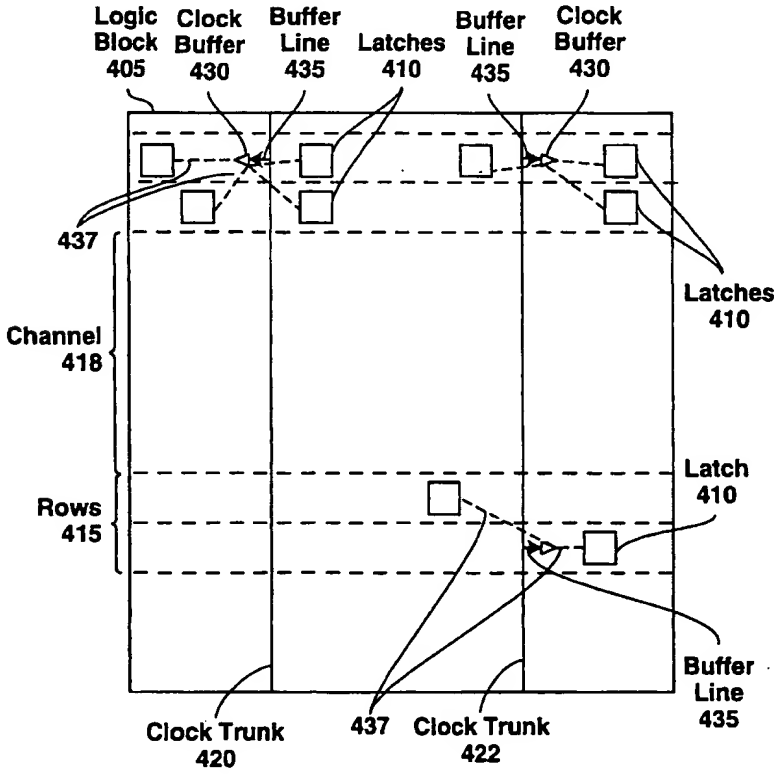
Primary Examiner—Krisna Lim

Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] ABSTRACT

A method and apparatus for inserting clock buffers to reduce clock skew in a logic block in which the proper placement of the cells within the logic block is first determined. Given this cell placement and the location of the local clock lines, the placement of clock buffers within the logic block is determined such that the clock buffers are in close proximity to the local clock lines. Routing is then performed to connect the clock buffers to their corresponding clock trunks and the cells requiring clock signals to their corresponding clock buffers. The performance of the logic block is then evaluated. If the performance does not satisfy a predetermined minimum threshold then the cells are modified to satisfy the minimum threshold, or come closer to attaining it. The clock buffers are removed, and the proper placement of the new cells within the logic block is determined. Given this new cell placement a new set of clock buffers is placed and a new routing is created. The performance is then re-evaluated and, if the minimum threshold still has not been attained, the above process is repeated.

24 Claims, 9 Drawing Sheets



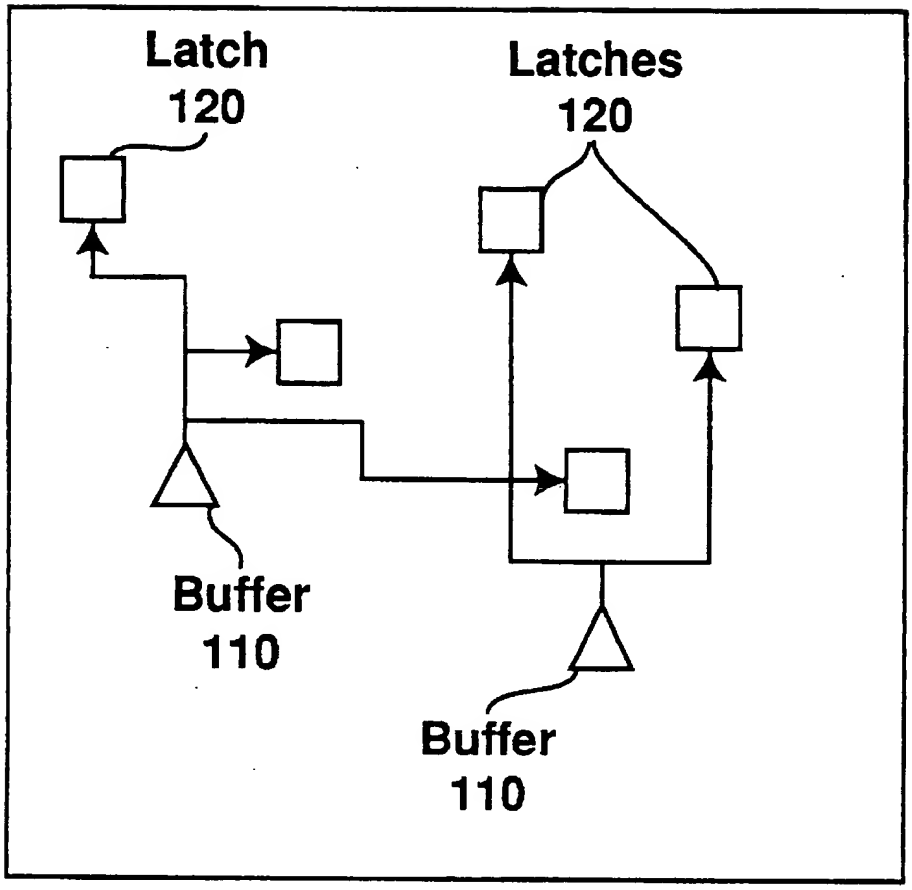


FIG. 1A
Prior Art

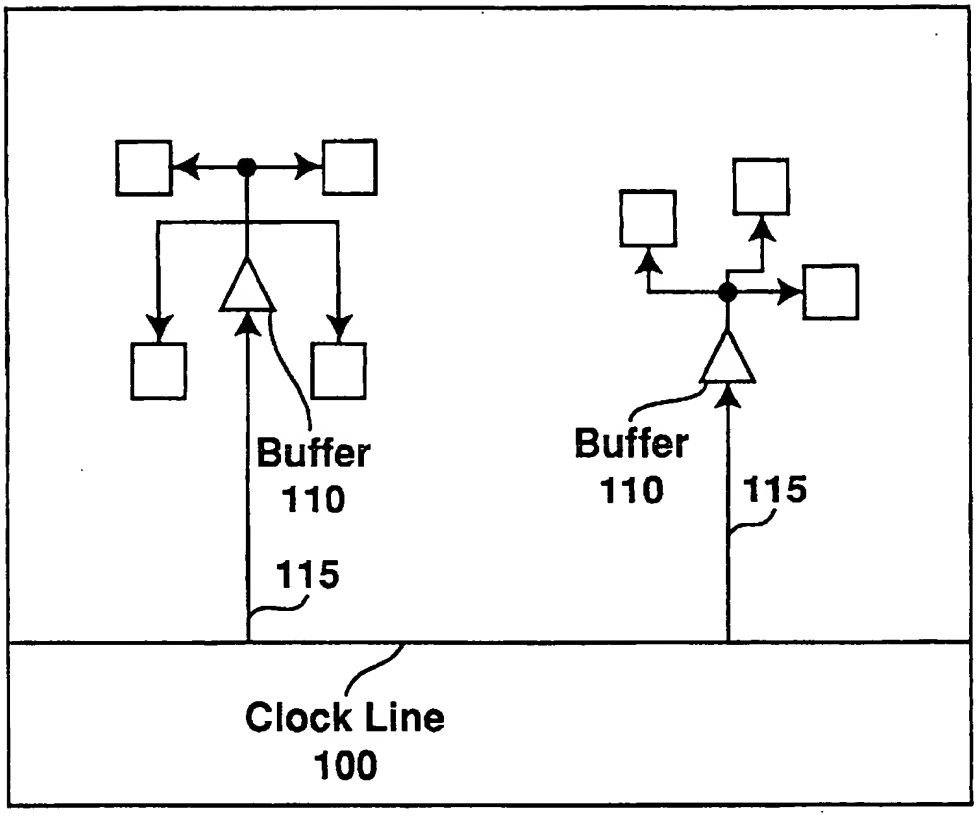


FIG. 1B
Prior Art

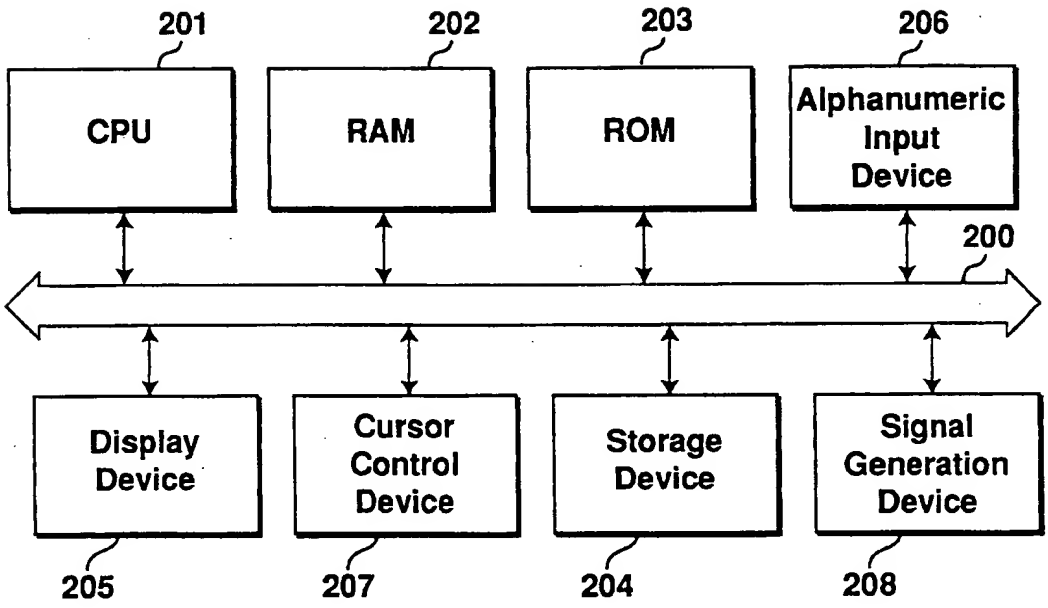

201
202
203
206
CPU
RAM
ROM
Alphanumeric Input Device
200
Display Device
Cursor Control Device
Storage Device
Signal Generation Device
205
207
204
208

FIG. 2

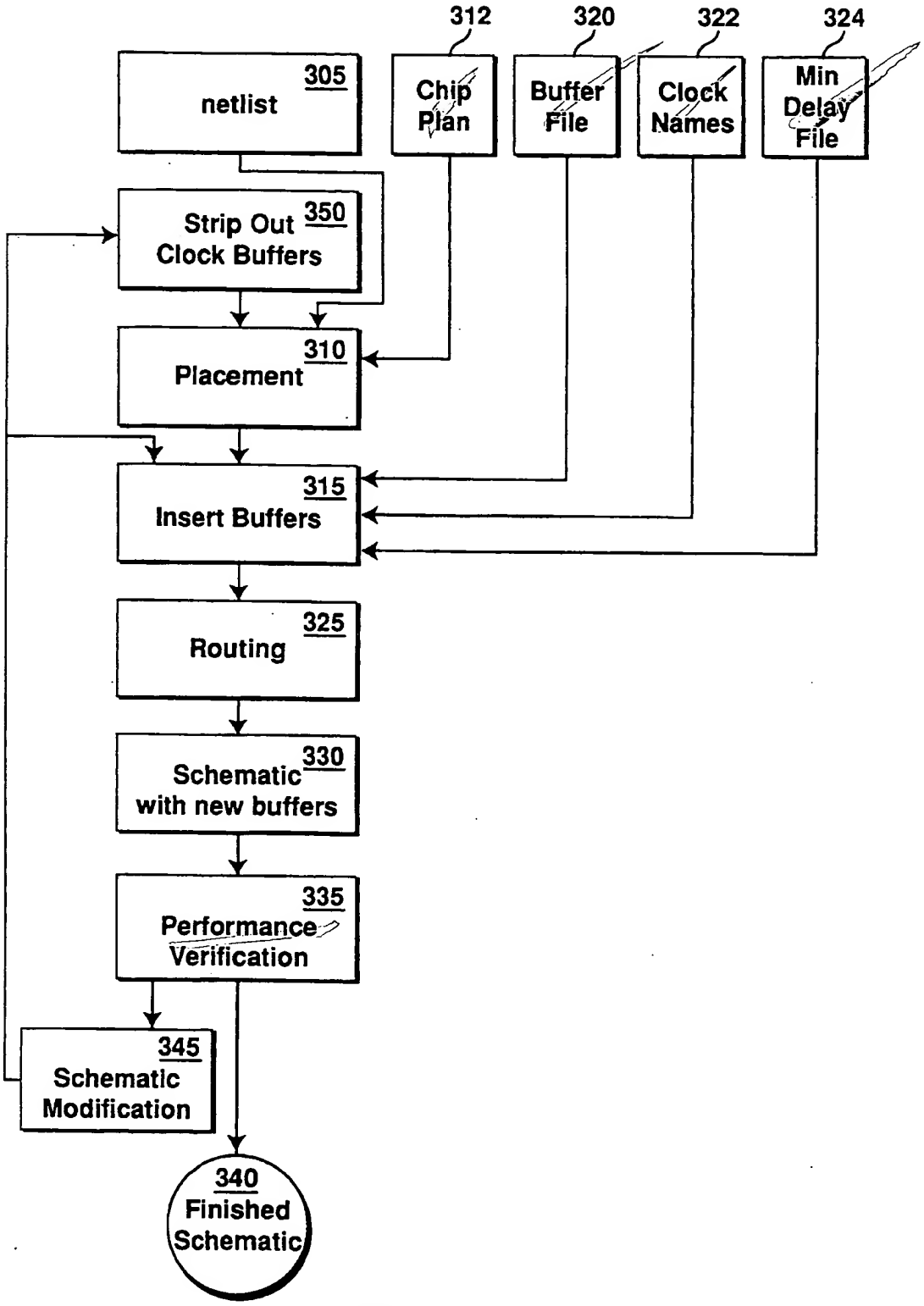

312
320
322
324
305
netlist
Chip Plan
Buffer File
Clock Names
Min Delay File
350
Strip Out Clock Buffers
310
Placement
315
Insert Buffers
325
Routing
330
Schematic with new buffers
335
Performance Verification
345
Schematic Modification
340
Finished Schematic

FIG. 3

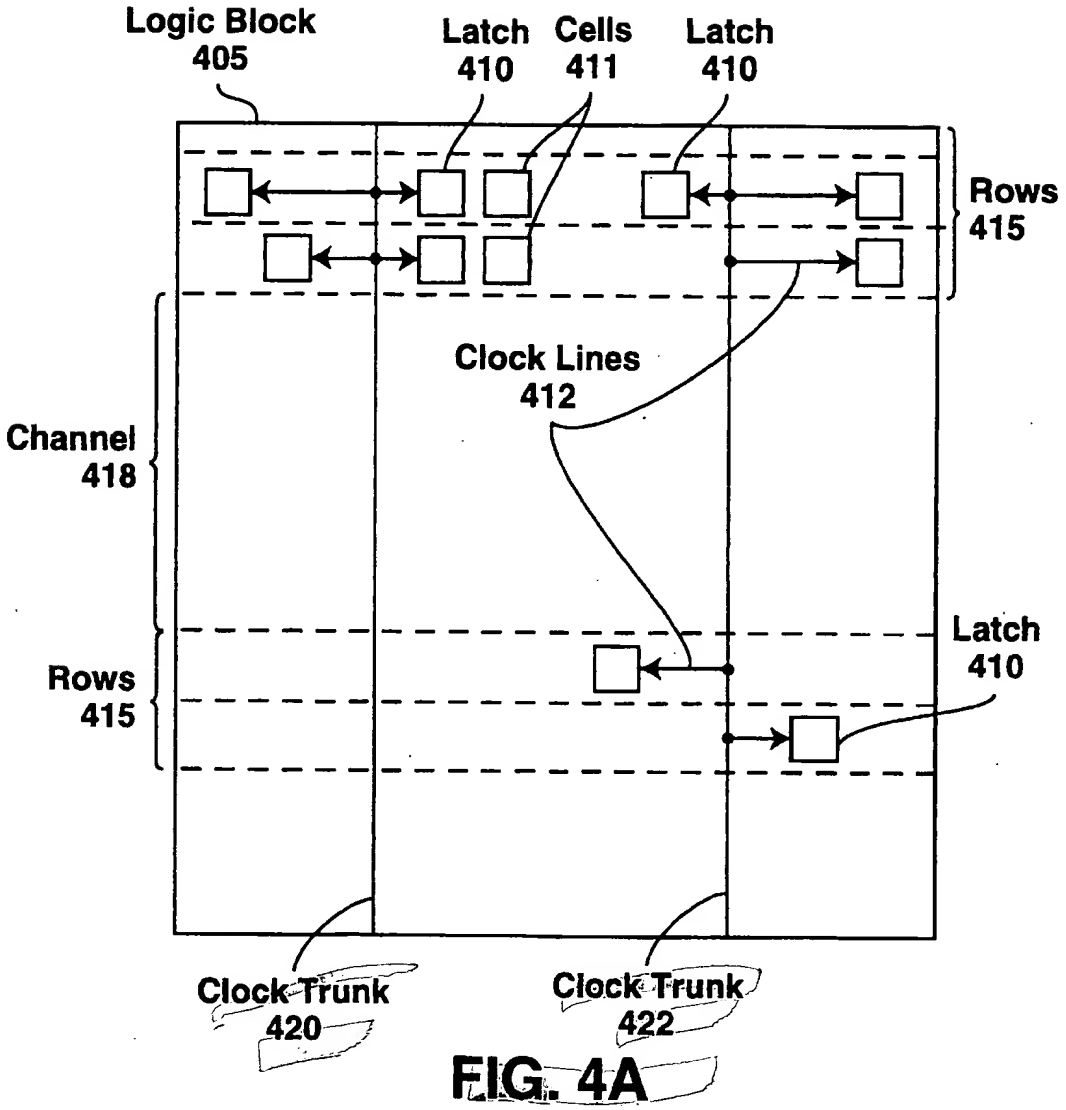

Logic Block
405
Latch
410
Cells
411
Latch
410
Rows
415
Clock Lines
412
Channel
418
Rows
415
Latch
410
Clock Trunk
420
Clock Trunk
422

FIG. 4A

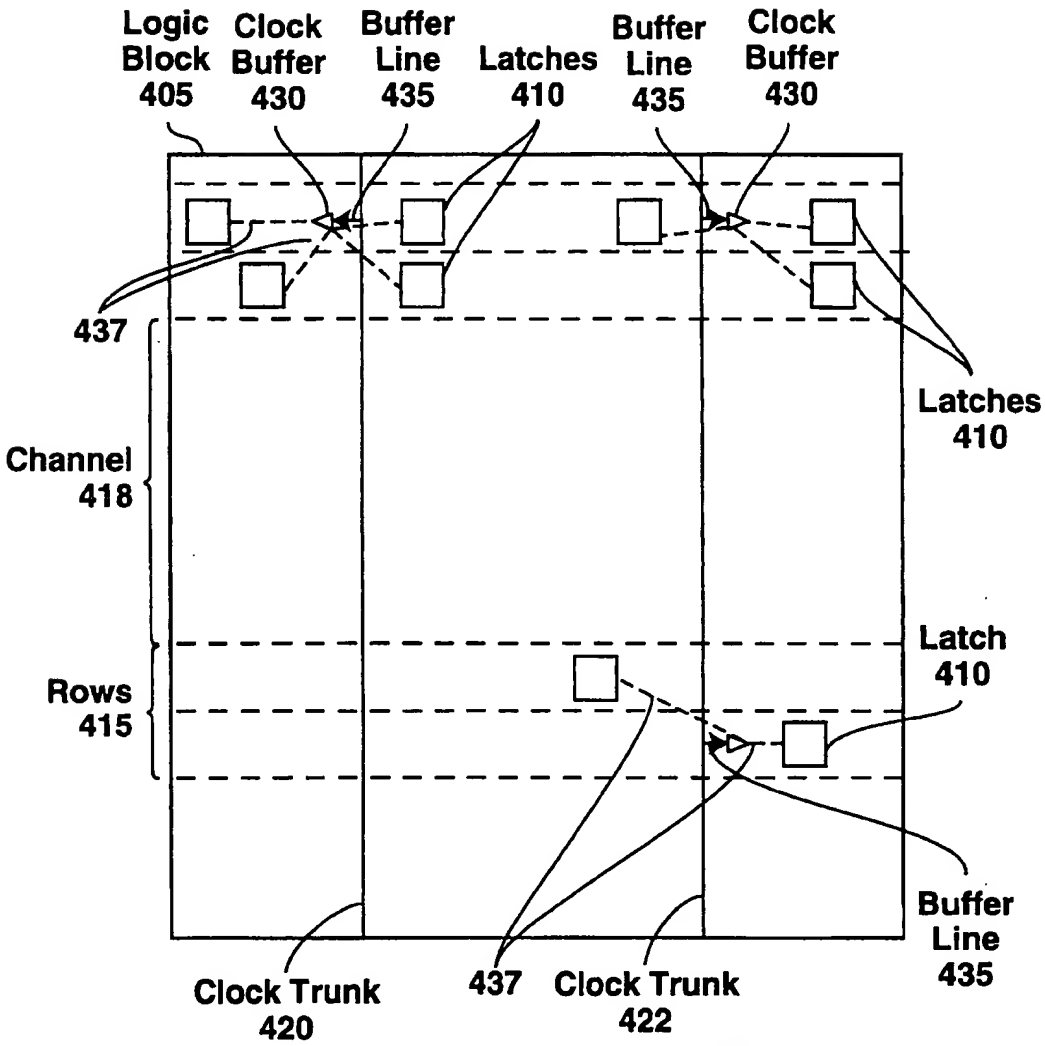

Logic Block 405
Clock Buffer 430
Buffer Line 435
Latches 410
Buffer Line 435
Clock Buffer 430
437
Channel 418
Rows 415
Latches 410
Latch 410
Buffer Line 435
Clock Trunk 420
437
Clock Trunk 422

FIG. 4B

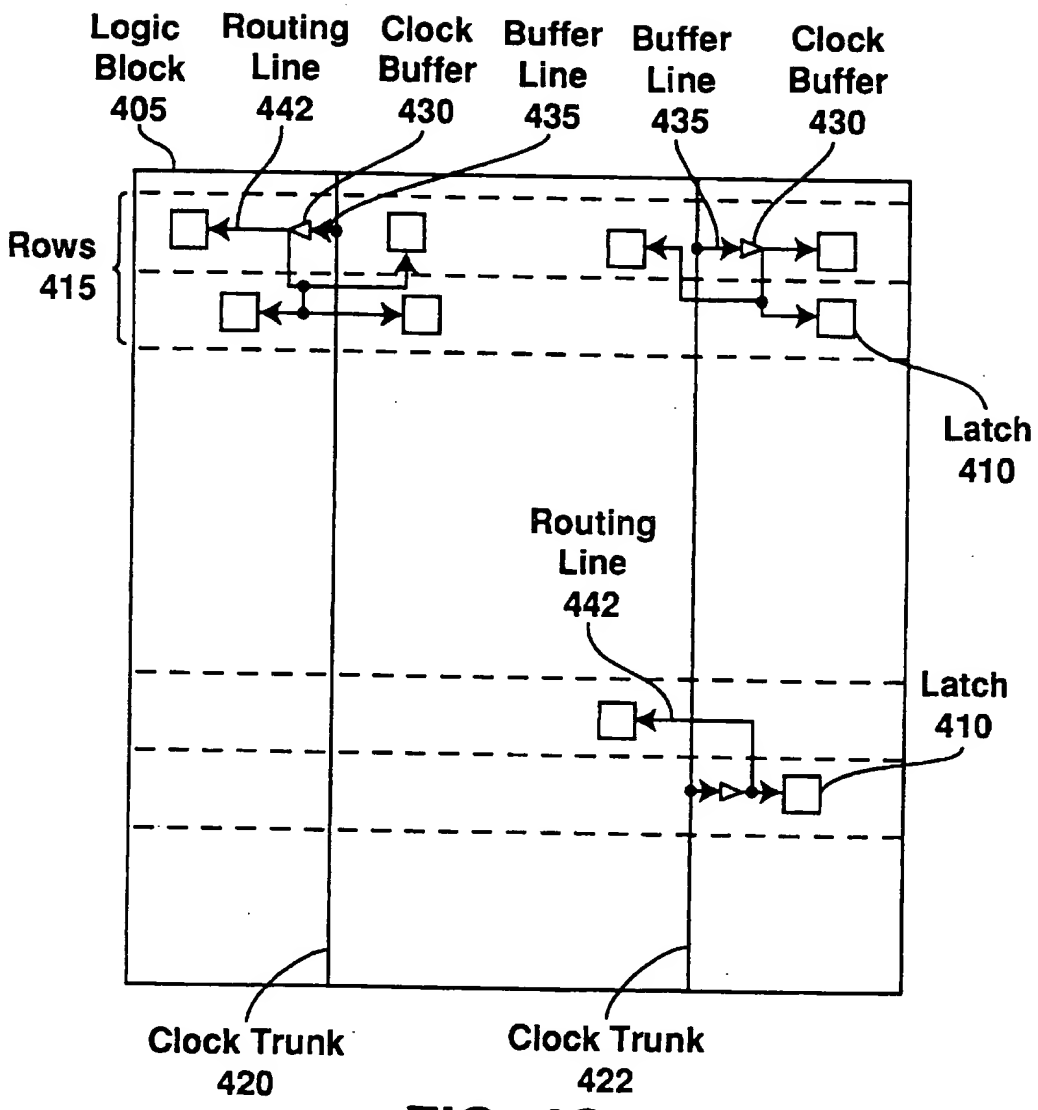

Logic Block 405
Routing Line 442
Clock Buffer 430
Buffer Line 435
Buffer Line 435
Clock Buffer 430
Rows 415
Latch 410
Routing Line 442
Latch 410
Clock Trunk 420
Clock Trunk 422

FIG. 4C

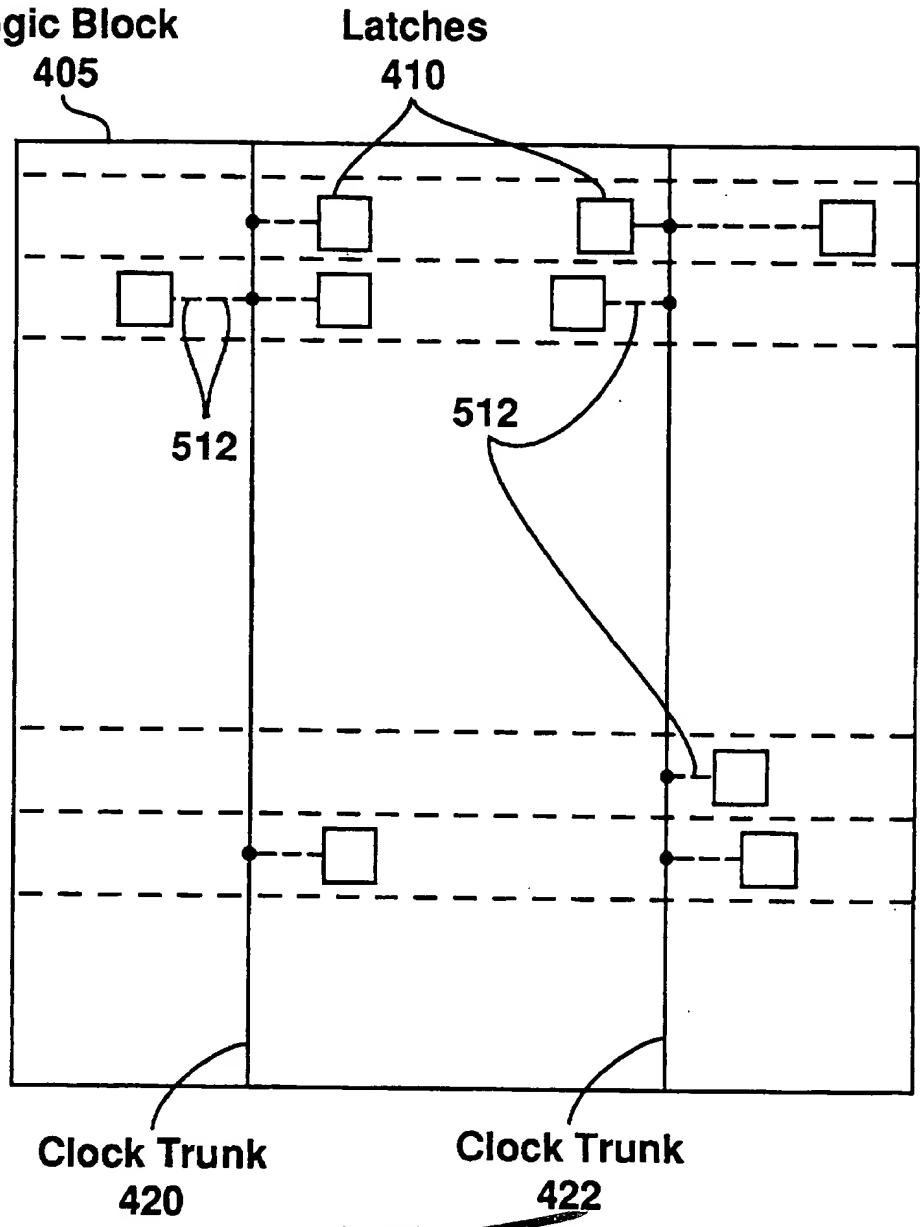

Logic Block
405
Latches
410
512
512
Clock Trunk
420
Clock Trunk
422

FIG. 5A

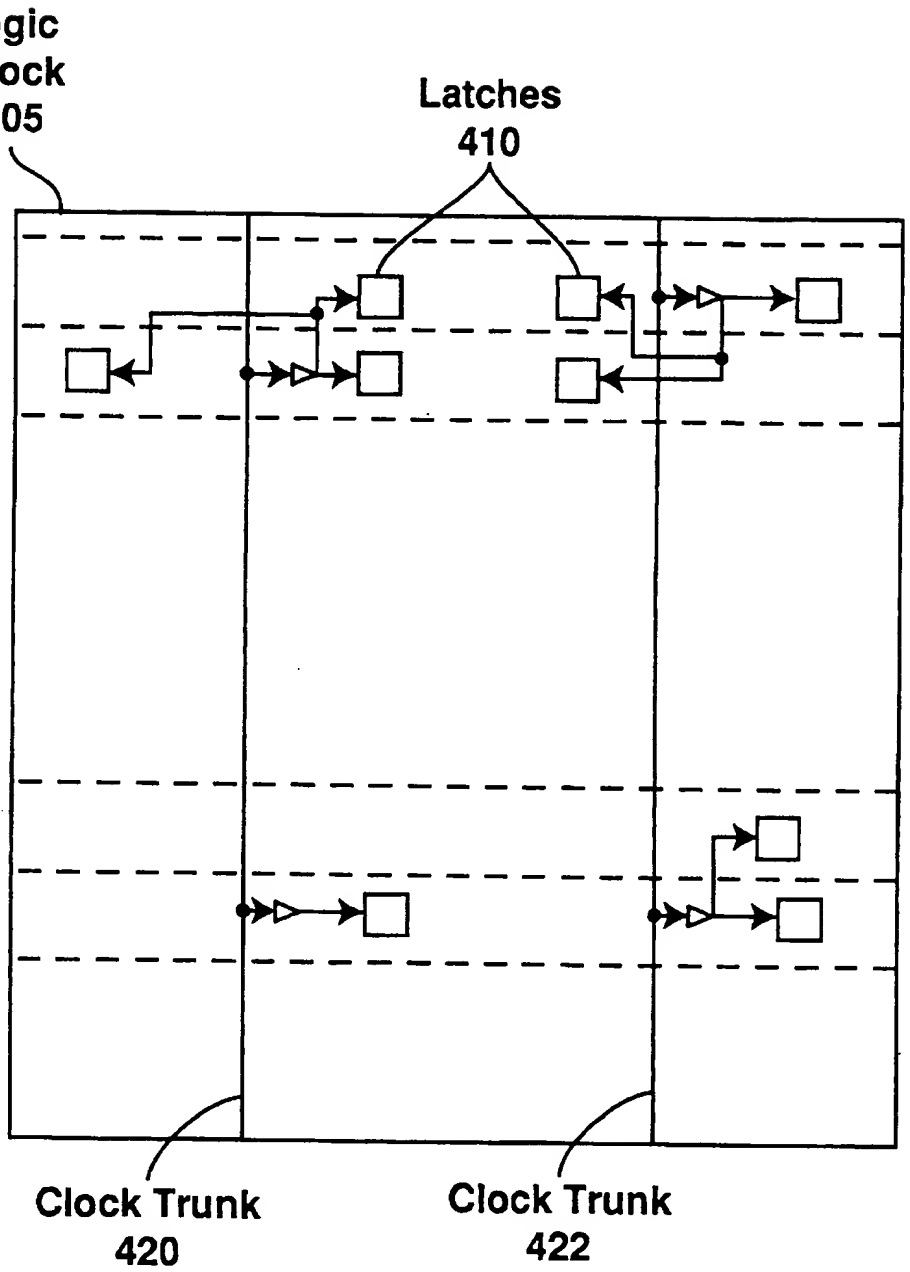


FIG. 5B

METHOD AND APPARATUS FOR AUTOMATICALLY INSERTING CLOCK BUFFERS INTO A LOGIC BLOCK TO REDUCE CLOCK SKEW

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention pertains to the field of microprocessor architecture and layout. More particularly, this invention relates to placing properly sized clock buffers in the proper location within a logic block to reduce clock skew.

2. Background

Components of an integrated circuit operate based on timing and pulsing of clock signals which provide a reference point or activation signal for circuit activity and processing. The clock signals also provide a timing or alignment reference which different circuits adopt when stepping through their respective processing tasks. It is important that the clocking signals be predictable and not delayed such that processing and execution by circuit components are accomplished in synchronization. Microprocessor integrated circuit devices utilize a system clock which provides timing and pulsing to drive the various elements and processing of the microprocessor.

It is vital to the operation of a microprocessor that the system clock be supplied uniformly to all components of the microprocessor with minimal clock skew. Clock skew refers to the variations in timing delays between a system clock and a clock signal reaching a component. Resistance within the clock line and capacitance on the clock line creates RC skews, a type of clock skew, as the clock signal propagates. Clock buffers can be used to deskew the clock signal, thus a system for automatically placing the proper clock buffers in the correct locations would be advantageous.

A similar problem is that of a minimum delay between latches. A minimum delay problem may arise when the signal from a source latch is input into a receiving latch. If the clock signal driving the receiving latch reaches the receiving latch after the signal from the source latch arrives, the receiving latch may latch the wrong data. Thus, a buffer may be inserted into the line between the two latches to create a delay such that the signal does not arrive prior to the clock signal.

Design techniques for microprocessors may include utilization of a large number of functional blocks in order to shorten the design cycle. The functional blocks consist of a varying number of cells and utilize clock buffers to prevent clock skew. As microprocessors use faster and faster clock speeds, variations in clock skew within the functional blocks becomes a major concern. The slower clock speeds used in older microprocessor technology were slow enough that the clock skew within the functional blocks could be either ignored or resolved easily. However, faster clock speeds require that the problem of clock skew be addressed more efficiently.

In addition, microprocessor development times have become shorter and shorter. Therefore, an automatic system for the designer to insert the properly sized clock buffer in the proper location would be advantageous.

Thus, it would be advantageous to automatically optimally insert the proper clock buffers into the functional blocks. The present invention offers such a solution.

An example prior art placement of clock buffers is shown in FIG. 1A. The clock buffers **110** may have been placed arbitrarily, or at the very least in a non-optimized manner.

That is, the clock buffers **110** were not guaranteed to be placed close to the clock line and thereby reduce clock skew. Additionally, as shown, the latches **120** were not necessarily driven by the clock buffer **110** located closest to each latch **120**.

Other prior art placements of clock buffers may have attempted to solve the clock skew problem by placing clock buffers close to the latches being driven, as shown in FIG. 1B. However, as is readily apparent in FIG. 1B, the placement of the clock buffers **110** is not optimized because the buffers **110** are not placed close to the clock line **100**. The extra distance between the clock line **100** and the buffers **110** over narrower lines **115** cause additional RC skew, thus, the reduction of clock skew is not optimized.

SUMMARY AND OBJECTS OF THE INVENTION

The present invention comprises a method and apparatus for inserting local clock buffers in a logic block. The present invention first determines the proper placement of the cells within the logic block. Then, given the cell placement and the location of the local clock trunks, the invention places the clock buffers within the logic block in close proximity to the local clock trunks. Routing is then performed to connect the clock buffers to their corresponding clock trunks and the cells requiring a clock signal to their corresponding clock buffers.

The performance of the block is then evaluated. If the performance does not meet a predetermined minimum threshold then the cells are modified to attain the minimum threshold, or come closer to attaining it. The clock buffers previously inserted are removed, and the proper placement of the new cells within the logic block is determined. Then, given this new cell placement and the location of the local clock trunks, the invention places a new set of clock buffers within the logic block in close proximity to the local clock trunks. A new routing is then created to connect the clock buffers to their corresponding clock trunks and the cells to their corresponding clock buffers.

The performance of this new block is then evaluated to determine whether it meets the predetermined minimum threshold. If it does, then the process is complete. However, if the minimum threshold is not satisfied then the system repeats the above process, modifying the cells, re-placing them, and re-inserting a new set of clock buffers. This process will be repeated until the minimum threshold is satisfied.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

FIG. **1A** is a block diagram of a prior art placement of clock buffers;

FIG. **1B** is a block diagram of a prior art placement of clock buffers;

FIG. **2** is a block diagram of a computer system used by the preferred embodiment of the present invention;

FIG. **3** is a flow chart of the steps of the preferred embodiment of the present invention;

FIG. **4A** is a diagram showing the placement of cells without clock buffers in the preferred embodiment of the present invention;

FIG. 4B is a diagram showing the placement of cells and clock buffers before routing in the preferred embodiment of the present invention;

FIG. 4C is a diagram showing the placement of cells and clock buffers after routing in the preferred embodiment of the present invention;

FIG. 5A is a diagram showing the placement of cells after having the clock buffers stripped in the preferred embodiment of the present invention; and

FIG. 5B is a diagram showing a new placement of cells and clock buffers in the preferred embodiment of the present invention.

DETAILED DESCRIPTION

In the following detailed description of the present invention numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one skilled in the art that the present invention may be practiced without these specific details. In other instances well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure the present invention.

Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

In general, computer systems used by the preferred embodiment of the present invention are as illustrated in block diagram format in FIG. 2, and comprise a bus **200** for communicating information, a central processor **201** coupled with the bus for processing information and instructions, a random access memory **202** coupled with the bus **200** for storing information and instructions for the central processor **201**, a read only memory **203** coupled with the bus **200** for storing static information and instructions for the processor **201**, a data storage device **204** such as a magnetic disk and disk drive coupled with the bus **200** for storing information (such as audio or voice data) and instructions, a display device **205** coupled to the bus **200** for displaying information to the computer user, an alphanumeric input device **206** including alphanumeric and function keys coupled to the bus **200** for communicating information and command selections to the central processor **201**, a cursor control device **207** coupled to the bus for communicating user input information and command selections to the central processor **201**, and a signal generating device **208** coupled to the bus **200** for communicating command selections to the processor **201**.

The display device **205** utilized with the computer system and the present invention may be a liquid crystal device, cathode ray tube, or other display device suitable for creating graphic images and alphanumeric characters (and ideographic character sets) recognizable to the user. The cursor control device **207** allows the computer user to dynamically signal the two dimensional movement of a visible symbol (pointer) on a display screen of the display device **205**. Many implementations of the cursor control device are known in the art including a trackball, mouse, joystick or special keys on the alphanumeric input device **205** capable of signaling movement of a given direction or manner of displacement. It is to be appreciated that the cursor means **207** also may be directed and/or activated via input from the keyboard using special keys and key sequence commands. Alternatively, the cursor may be directed and/or activated via input from a number of specially adapted cursor directing devices, including those uniquely developed for the disabled. In the discussions regarding cursor movement and/or activation within the preferred embodiment, it is to be assumed that the input cursor directing device or push button may consist of any of those described above and specifically is not limited to the mouse cursor device.

Logic blocks comprising a plurality of standard cells are well known in the art. In the preferred embodiment of the present invention, these cells are placed and the proper clock buffers to drive the latches (or any other cells requiring a clock signal) are automatically inserted; the block is then modified, if necessary, to satisfy performance requirements.

In the preferred embodiment, the cells are placed within the logic block in a double back row configuration. That is, cells are placed in pairs of rows **415** separated by channels **418**, as shown in FIG. 4A. However, it should be readily apparent to those of ordinary skill in the art that the present invention may be utilized in any of a wide variety of placement configurations.

A flowchart of the method of the preferred embodiment of the present invention is shown in FIG. 3. A schematic, or netlist, is first created, step **305**, which describes the connectivity between the cells within the block. In a logic block with multiple clock signals the schematic also describes which clock signal should be connected to, and therefore drive, which latches. If a particular logic block utilized only a single clock signal, then the schematic would describe each latch as being connected to that single clock signal. The schematic also contains the loads of each latch within the schematic, which is described in more detail below.

The schematic defines the connectivity between cells and clock signals, however, it does not describe the placement of the cells in relation to one another or the clock trunks. In the preferred embodiment of the present invention, a netlist is used to describe the connectivity, however it should be apparent to those of ordinary skill in the art that any description of the cell connectivity could be used.

The netlist from step **305** is input into a standard cell placement system, step **310**. The cell placement system

places the cells within the row-pairs to optimize the routing and the area of the logic block. Thus, the cell placement system places the cells to obtain the best connectivity between the cells. In the preferred embodiment of the present invention, the Timberwolf placement algorithm is used by the placement system in step **310**. It should be apparent to those of ordinary skill in the art, however, that any standard cell placement system may be utilized.

In the preferred embodiment, an additional input to the placement system is the chip plan **312**. The chip plan **312** is a description of the pin layout of the chip containing the logic block being designed. One element of the chip plan **312** is a description of the local clock line(s) which drive the cells within the logic blocks on the chip. Thus, the chip plan **312** identifies to the placement system the location(s) of the clock line(s) used by any particular logic block.

An example placement of cells is shown in FIG. 4A. Multiple latches **410** are shown placed within multiple double back rows **415** in a logic block **405**. Additional cells **411** which do not require a clock signal are also shown. A channel **418**, containing no cells, is located between each row pair **415**. Only two row pairs **415** are shown in FIG. 4A to avoid unnecessarily cluttering the drawing. It should be apparent to those of ordinary skill in the art that the number of row pairs **415** within a block could vary among blocks, with the maximum limit being based on the size of the block **405**.

Multiple clock trunks **420** and **422** are also shown in FIG. 4A. Local clock trunks **420** and **422** may be driven by the same (system) clock or by separate clocks. The location of these trunks **420** and **422** are provided by the chip plan **312**, described above.

The placement system, step **310** of FIG. **3**, places the cells within the row pairs as described above. If the standard routing were performed, without the insertion of clock buffers, the latches would be connected directly to the clock trunks over clock lines **412**, as shown in FIG. 4A. Such a placement creates a substantial clock skew problem as a result of RC skew due to the narrowness of clock lines **412** and the differences in loads being driven. Furthermore, the prior art placement of clock buffers would not optimally solve the clock skew problem, as discussed above.

After the placement of the cells is completed, the proper clock buffers are inserted, step **315**. The clock buffers are inserted based on the placement of the cells determined in step **310**, the information in the clock buffer file **320**, and the information in the clock names file **322**.

The clock names file **322** lists the names of the separate clock signals which are utilized in the logic block **405**. For example, FIG. **4A** shows a logic block **405** having two clock trunks **420** and **422**. Clock trunks **420** and **422** could carry the same clock signal, e.g. CLOCKA. Alternatively, clock trunks **420** and **422** may carry separate clock signals, e.g. clock line **420** may carry CLOCKA while clock trunk **422** may carry CLOCKB. The location of these clock trunks is described in the chip plan **312**. The clock signal on each clock trunk is described in the clock names file **322**.

The clock buffer file **320** contains a description of the clock buffers available for use by the present invention. A wide variety of commercially available buffers exist which could be utilized by the present invention. Each clock buffer which could be used has a C_{min} and a C_{max} value associated with it. The C_{min} value is the minimum capacitance load the buffer is capable of driving, whereas C_{max} is the maximum load the buffer is capable of driving. Thus, the clock buffer file **320** contains a description of the clock buffers available and their respective C_{min} and C_{max} values. Table 1 shows the data contained within an example clock buffer file.

TABLE 1

Buffer	C_{min}	C_{max}
Buffer A	0.079	0.115
Buffer B	0.115	0.170
Buffer C	0.170	0.255
Buffer D	0.255	0.380
Buffer E	0.380	0.566
Buffer F	0.566	0.851
Buffer G	0.851	1.274
Buffer H	1.274	1.903

The present invention determines the proper size and number of clock buffers to be used based on the clock buffers described in clock buffer file **320** and the load to be driven by the buffer(s) within a row pair. The load to be driven by the buffer(s) is based on the load of the latches **410** to be driven and the load of the line connecting the buffer(s) to the latches **410**. These loads were received as inputs to the placement system **310**. That is, the original netlist from step **305** contains the loads of each latch contained therein. This information is contained in an auxiliary file along with the netlist in the preferred embodiment. The load of the line is determined based on the length of the line and its load per unit length. The load per unit length may be input separately at the clock buffer insertion step, or may be input from the placement system along with the latch loads. The length of each line can be determined given the location of the clock trunks **420** and **422** and the placement of the cells in step **310**.

It should be noted that in the preferred embodiment of the present invention a minimum threshold exists below which the load of a particular line need not be considered. The main load being driven by a clock buffer in the present invention is the latches. Thus, if a line between a buffer and a clock trunk or latch is short enough, the load will be insignificant relative to the load of the latches and can safely be ignored.

In the currently preferred embodiment of the present invention, the load of a line may be safely ignored if the length is less than 150 microns. Furthermore, in the currently preferred embodiment, the chip plan **312** has the clock trunks **420** and **422** located no more than 300 microns apart. Thus, the length of the lines connecting the clock buffers and the latches **410** may be safely ignored.

Given the loads to be driven by the clock buffer(s) and the load each buffer is capable of driving, the present invention matches these values and determines the proper size and number of buffers required to drive the load. In the currently preferred embodiment of the present invention, a single buffer will be used to drive the load. If a single buffer large enough to drive the load is not available in clock buffer file **320** then an additional buffer(s) will be added. For example, suppose the clock buffer file **320** contained two buffers, BUF1 having a C_{min} of 0.566pF and C_{max} of 0.851pF, and BUF2 having a C_{min} of 0.851pF and a C_{max} of 1.274pF. Further suppose that the load to be driven by the buffer(s) was 1.900pF. The present invention would compare these values and determine that two buffers are required to drive the load: a BUF1 and a BUF2.

The currently preferred embodiment utilizes a single buffer to drive the load if one of an appropriate size is available. However, it should be readily apparent to those of ordinary skill in the art that multiple smaller buffers may be used rather than a single larger buffer.

Having determined the size and number of clock buffers to use, the currently preferred embodiment of the present

invention places the buffers in the row pairs **415** and modifies the netlist such that the latches receive clock signals from a buffer in the row pair rather than directly from the clock trunks **420** or **422**. The present invention places the clock buffers as close as possible to the clock trunk **420** or **422** which is closest to the latch being driven. That is, the present invention places the clock buffers in the available position within the row pairs **415** which is closest to the clock trunks **420** and **422**. Alternatively, the present invention may place the clock buffers right below and very close to the clock trunks **420** and **422**. Thus, the distance between the clock trunk and the clock buffer is minimized. The present invention modifies the netlist to include the clock buffers in these closest available positions to the clock trunks.

The placement of clock buffers is performed within each row pair **415** for each clock trunk **420** and **422**. Thus, each clock trunk **420** and **422** will have at least one clock buffer **430** placed close to it in each row pair which has a cluster of latches (or single latch) to be driven.

FIG. **4B** shows the placement of clock buffers **430** within row pairs **415** of a logic block **405**. Note that the buffer lines **435** will not actually be placed until routing step **435**. Clock trunks **420** and **422** are wider than buffer lines **435** will be, therefore they cause a lower RC delay (and therefore result in less clock skew). When the clock signals travel from a clock trunk **420** or **422** to a clock buffer **430** over buffer line **435**, the RC delay is greater because the lines **435** are narrower. Thus, by placing the buffers **430** very close to the clock trunks **420** and **422** the length of the lines **435** is reduced, and the RC delay attributable to the lines **435** is minimized.

In the currently preferred embodiment of the present invention the RC delay caused by the lines **435** is below the minimum threshold, discussed above. Thus, the load attributable to the buffer wires **435** may be safely disregarded. The load attributable to the buffer wires **435** will be below the minimum threshold because the clock buffers are inserted close to the clock trunks. Note that under certain circumstances the row pairs could be full of cells such that no place is available for a clock buffer. In this situation, the present invention may reposition the cells relative to the clock trunks in order to create a position for the clock buffer. This repositioning, however, could result in a cell being pushed outside of the functional block boundary. In such a situation, the placement of the cells, step **310**, must be repeated, or alternatively the chip plan **312** must be modified to increase the size of the functional block.

Note that in some instances multiple clock buffers **430** may be inserted into a row pair to drive multiple latches **410** within that row pair. In this situation the present invention must determine which latches are driven by which buffers. The preferred embodiment of the present invention resolves this situation by determining the latches closest to each buffer, as determined by the modified netlist. The present invention will drive latches **410** with the buffer **430** closest to each latch **410**, limited by the loads being driven and the C_{min} and C_{max} of each buffer, described above.

A related situation is determining which buffer drives a latch located between two clock trunks. In the preferred embodiment of the present invention the buffer which is closest to the latch in question will drive the latch. However, it should be readily apparent to those of ordinary skill in the art that other solutions may exist, such as having a buffer with extra driving capacity drive the latch even though it may not be closest.

FIG. **4B** shows a logic block **405** after insertion of the clock buffers **430**. Dashed lines **437** show the modified connections between the latches and the clock buffers **430**. As shown, the latches **410** are no longer directly connected to the clock trunks **420** and **422**. The latches **410** receive clock inputs from the clock buffers **430**, as described above, which in turn are connected to the clock trunks **420** and **422** over buffer lines **435**.

It should be noted that the rare situation may occur in which a single latch is placed in a row pair having a load smaller than the smallest C_{min} in the clock buffer file **320**. In the preferred embodiment of the present invention, this problem is resolved by repeating step **310**. That is, the placement system replaces the cells repeatedly until no such single latch remains. Alternatively, the designer may manually place an existing clock buffer of the appropriate size to drive that single latch.

The preferred embodiment of the present invention determines the proper routing for the block, step **325**, after completing the insertion of the clock buffers. The routing system takes the modified netlist and determines the best routing between the clock buffers **430** and the latches **410**, each buffer is driving. Routing systems are well known in the art and any commercially available routing system may be utilized to perform the routing step **325**.

Upon completion of routing step **325**, a new schematic is produced, step **330**, having all cells and buffers placed and the routing completed. An example schematic is shown in FIG. **4C**, showing the routing lines **442** placed in routing step **325**.

The new schematic is then input into an analysis system, step **335**, which performs a performance verification of the logic block **405**. In analysis step **335** the performance of the block **405** is compared to a minimum performance threshold defined by the system designer. This threshold is typically the minimum speed at which block **405** must run in order to function properly within the environment the block **405** is to be placed. Such analysis systems are well known in the art and thus will not be discussed further.

Upon completion of the analysis step **335**, one of two actions may be taken. If the minimum performance threshold was satisfied then the present invention outputs a finished schematic, step **340**. However, if the minimum threshold is not satisfied, then the schematic is further modified to attain or approach the minimum performance threshold, step **345**.

The analysis system which performed the timing analysis in step **335** is also capable of modifying the cells within the logic block **405**. A wide variety of optimization techniques exist, such as upsizing gates to make signals faster. Any of a wide variety of optimization techniques known to those of ordinary skill in the art may be utilized to attain or approach the minimum timing threshold. It should be readily apparent to those of ordinary skill in the art that although these optimization techniques are designed to attain or approach a minimum timing threshold, under certain circumstances a particular technique may modify cells such that timing is not improved.

These optimization techniques, however, may affect the placement of the cells within the row pairs. For example, after modification the cells may be too large to fit within the row pairs as they previously did. Thus, the cells in block **405** must be replaced by the placement system, as done in step **310**. Before this can be accomplished, however, the clock buffers **430** which were inserted in step **315** must be removed from the netlist because their placement will no

longer be optimal due to the re-placement to be performed, in step **310**.

The clock buffers **430**, previously placed in step **315**, are removed in step **350**. This removal is automatic; that is, no user intervention is required. The present invention further modifies the netlist from step **345** by eliminating the clock buffers; the netlist is also modified such that the latches, which previously received clock signals from a clock buffer, now receive a clock signal directly from a clock trunk **420** or **422**, as shown in FIG. **5**A. The dashed lines **512** show which signals drive which latches, however the actual routing has not occurred yet. It should also be noted that the location of the cells shown in FIG. **5**A has no significance as the cells have not been placed yet.

After removal of the clock buffers, the schematic, without the buffers, is again input into the placement system, step **310**. The placement system, described above, will re-place the modified cells as described above. The present invention will repeat steps **315** through **325** as described above, producing a new schematic at step **330**. An example of a new schematic is shown in FIG. **5**B.

The timing analysis, step **335**, will be performed again. As described above, if the new schematic satisfies the performance requirements then the present invention will output a finished schematic, step **340**. If the performance requirements are still not satisfied, the cells will again be modified, the clock buffers removed, and the above process repeated. The above process will be repeated until the performance requirements are satisfied.

In an alternate embodiment, an additional buffer may be inserted between two latches after the performance verification of step **335**. The performance verification may determine that a minimum delay problem exists, that is, the signal from one latch may be arriving at the receiving latch prior to the arrival of the clock signal. If this minimum delay problem is the only remaining problem, it may be desirable to not re-place the cells again. Thus, a buffer may be inserted after the performance verification to delay the signal and avoid latching the wrong data into the receiving latch.

Thus, the buffer is inserted, step **315**. The proper size buffer to insert is input at step **315** and is selected based on the minimum delay required, as determined at performance verification, step **335**. That is, the proper size buffer will generate a delay greater than the minimum delay required. A minimum delay file **324** contains a set of minimum delays which must be resolved and the latches or elements each minimum delay is associated with, and contains buffers to solve each minimum delay problem.

Given the size of the buffer needed and the latch which has the minimum delay problem, the present invention inserts the proper buffer. Since the buffer resolves the minimum delay problem between the two latches, it may be placed anywhere along the signal path from the source latch to the receiving latch. According to the present invention, buffers are placed in any available location along the signal path.

Routing, step **325**, is then repeated. The placement of existing cells was not modified, thus the routing will change minimally. Upon completion of the routing a new schematic is generated, step **330**, and the performance verification repeated, step **335**. The insertion of the buffers to solve the minimum delay problems should resolve the remaining timing concerns, thus a finished schematic will be output, step **340**.

It should be noted that if another minimum delay problem arises, steps **315** through **340** may be repeated at a future date. In such a situation, a buffer which resolves the minimum delay is inserted, step **315**, routing takes place as discussed above, and a new finished schematic is produced, step **340**.

The preferred embodiment of the present invention, a method and apparatus for inserting clock buffers, is thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.

What is claimed is:

1. A method for automatically reducing clock skew in a logic block having a plurality of cells, the method comprising the steps of:

(a) determining a placement of said plurality of cells within said logic block;

(b) determining a placement of a plurality of clock buffers within said logic block such that each clock buffer of said plurality of clock buffers is located in close proximity to a clock line; and

(c) determining a routing between said plurality of clock buffers, said plurality of cells and said clock line.

2. A method for reducing clock skew as claimed in claim **1** further comprising the step of:

(d) determining the performance of said logic block.

3. A method for reducing clock skew as claimed in claim **2** further comprising the steps of:

(e) removing said plurality of clock buffers from said logic block if said performance is below a predetermined minimum threshold;

(f) modifying at least one cell of said plurality of cells if said performance is below said minimum threshold; and

(g) repeating steps (a) through (g) if said performance is below said minimum threshold.

4. A method for reducing clock skew as claimed in claim **1** wherein said step of determining a placement of a plurality of clock buffers comprises locating each clock buffer of said plurality of clock buffers in a closest available position to a clock line.

5. A method for reducing clock skew as claimed in claim **1** wherein said step of determining a placement of said plurality of cells comprises receiving a description of connectivity between a plurality of cells.

6. A method for reducing clock skew as claimed in claim **1** wherein said step of determining a placement of a plurality of clock buffers further comprises receiving a description of the location of said clock line.

7. A method for reducing clock skew as claimed in claim **1** wherein said step of determining a placement of said plurality of cells comprises placing said cells within a plurality of rows.

8. A method for reducing clock skew as claimed in claim **1** wherein said step of determining the placement of a plurality of clock buffers further comprises placing a first set of clock buffers of a first clock buffer type corresponding to a first clock and placing a second set of clock buffers of a second clock buffer type corresponding to a second clock.

9. A method for reducing clock skew as claimed in claim **1** wherein said step of determining the placement of said plurality of clock buffers further comprises placing a first set of clock buffers corresponding to a first clock line and placing a second clock buffer corresponding to a second clock line.

10. A method for reducing clock skew as claimed in claim **1** wherein said step of determining said routing comprises

coupling each buffer of said plurality of buffers to said clock line and coupling each cell of said plurality of cells requiring a clock signal to a buffer of said plurality of buffers such that no cell requiring a clock signal is directly coupled to said clock line.

11. A method for automatically inserting buffers into a logic block having a plurality of cells, the method comprising the steps of:

(a) determining a placement of a plurality of cells within said logic block;

(b) determining a placement of a plurality of buffers within said logic block;

(c) determining a routing between said plurality of buffers, said plurality of cells and said clock line;

(d) determining the performance of said logic block;

(e) removing said plurality of buffers from said logic block if said performance is below a predetermined minimum threshold;

(f) modifying a cell of said plurality of cells if said performance is below said minimum threshold; and

(g) repeating steps (a) through (g) if said performance is below said minimum threshold.

12. The method of claim **11** wherein said step (b) comprises determining a placement of a plurality of buffers within said logic block such that each clock buffer of said plurality of buffers is located in close proximity to a clock line.

13. The method of claim **11** wherein said step (b) comprises locating each buffer of said plurality of buffers in the closest available position to a clock line.

14. The method of claim **11** wherein said step (a) comprises placing said cells within a plurality of rows.

15. The method of claim **11** wherein said step (b) comprises placing a first buffer of a first buffer type and placing a second buffer of a second buffer type.

16. The method of claim **11** wherein said step (b) comprises placing a first buffer corresponding to a first clock line and placing a second clock buffer corresponding to a second clock line.

17. The method of claim **11** wherein said step (c) comprises coupling each buffer of said plurality of buffers to said clock line and coupling each cell of said plurality of cells requiring a clock signal to a buffer of said plurality of buffers such that no cell requiring a clock signal is directly coupled to said clock line.

18. The method of claim **11** wherein said step (g) comprises determining a placement of a buffer and repeating steps (c) through (g).

19. An apparatus for inserting clock buffers into a logic block having a plurality of cells to reduce clock skew, the apparatus comprising:

a bus;

a memory device which stores a set of available clock buffers, wherein the memory device is coupled to the bus; and

a processor, coupled to the bus, for

determining a placement of a plurality of cells within said logic block,

determining a placement of a plurality of clock buffers selected from the set of available clock buffers within said logic block such that each clock buffer of said plurality of clock buffers is located in close proximity to a clock line,

determining the routing between said plurality of clock buffers, said plurality of cells and said clock line,

determining the performance of said logic block,

removing said plurality of clock buffers from said logic block if said performance is below a predetermined minimum threshold, and

modifying a cell of said plurality of cells if said performance is below said minimum threshold.

20. An apparatus for inserting clock buffers as claimed in claim **19** wherein said processor for determining a placement of a plurality of clock buffers is also for locating each clock buffer of said plurality of clock buffers in a closest available position to a clock line.

21. An apparatus for inserting clock buffers as claimed in claim **19** wherein said processor for determining the placement of said plurality of cells places said cells within a plurality of rows.

22. An apparatus for inserting clock buffers as claimed in claim **19** wherein said processor for determining the placement of said plurality of clock buffers is also for placing a first set of clock buffers of a first clock buffer type corresponding to a first clock and placing a second set of clock buffers of a second clock buffer type corresponding to a second clock.

23. An apparatus for inserting clock buffers as claimed in claim **19** wherein said processor for determining the placement of said plurality of clock buffers is also for placing a first set of clock buffers corresponding to a first clock line and placing a second clock buffer corresponding to a second clock line.

24. An apparatus for inserting clock buffers as claimed in claim **19** wherein said processor for determining said routing is also for coupling each buffer of said plurality of buffers to said clock line and coupling each cell of said plurality of cells which requires a clock signal to a buffer of said plurality of buffers such that no cell requiring a clock signal is directly coupled to said clock line.

* * * * *